

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA

COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DISSERTAÇÃO DE MESTRADO

**FLOAPP: AMBIENTE INTEGRADO PARA SUPORTE ÀS
COMUNIDADES DE DESENVOLVIMENTO E USUÁRIAS DE
SOFTWARE LIVRE**

MESTRANDA

ANA ISABELLA MUNIZ

ORIENTADORA

FRANCILENE PROCÓPIO GARCIA

CAMPINA GRANDE

AGOSTO – 2007

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**FLOAPP: AMBIENTE INTEGRADO PARA SUPORTE ÀS
COMUNIDADES DE DESENVOLVIMENTO E USUÁRIAS DE
SOFTWARE LIVRE**

ANA ISABELLA MUNIZ

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande, como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

ÁREA DE CONCENTRAÇÃO: CIÊNCIA DA COMPUTAÇÃO

LINHA DE PESQUISA: ENGENHARIA DE SOFTWARE

ORIENTADORA
FRANCILENE PROCÓPIO GARCIA

CAMPINA GRANDE
AGOSTO – 2007

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

M966d

2007 Muniz, Ana Isabella.

FLOApp: ambiente integrado para suporte às comunidades de desenvolvimento e usuárias de software livre /Ana Isabella Muniz. — Campina Grande: 2007.

98f. : il

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

Referências.

Orientadora: Dr^a. Francilene Procópio Garcia.

1. Software Livre. 2. Ambiente Colaborativo. 3. Modelo de Negócio.

I. Título.

CDU 004.055 (043)

**"Fl.OApp: AMBIENTE INTEGRADO PARA SUPORTE ÀS COMUNIDADES
DE DESENVOLVIMENTO E USUÁRIAS DE SOFTWARE LIVRE"**

ANA ISABELLA MUNIZ

DISSERTAÇÃO APROVADA EM 29.08.2007


PROF.^a FRANCILENE PROCÓPIO GARCIA, D.Sc
Orientadora


PROF. JOSÉ ANTÃO BELTRÃO MOURA, Ph.D
Examinador


PROF. CLENIO FIGUEIREDO SALVIANO, Dr.
Examinador

CAMPINA GRANDE – PB

*“Há pessoas que transformam o sol numa simples mancha amarela,
mas há também aquelas que fazem de uma simples mancha amarela o próprio sol!”*

[Pablo Picasso]

AGRADECIMENTOS

A Deus, por estar sempre ao meu lado e me fazendo compreender “...minhas provações não são maiores que meu DEUS!!!”.

Aos meus pais, referência para toda uma vida!!

A minha tia Penha, seu esforço e dedicação contribuíram para o meu desenvolvimento profissional e pessoal.

Aos meus irmãos Alexandre, Patrícia, Tiago, Rafael, Junior e João Paulo pelo carinho, paciência para comigo em todos os momentos. Nossas risadas, tristezas e (...) estão guardados e nos momentos mais difíceis me ajudaram a levantar a cabeça e começar tudo de novo. Amo vocês!!!

Ao meu amor, Fábio.

Aos meus sobrinhos Jonas Gabriel, Denis Willian e Bianca, minha fonte de vida.

Ao meu cunhado Sandro, um grande amigo e irmão que Deus colocou de presente no meu caminho.

A minha cunhada Mércia, muito obrigada pela amizade!

Ao Lion (Carlos Alexandre) e o pixote (Guga) muito mais que amigos, co-orientadores, o melhor fruto deste mestrado.

A minha orientadora Francilene, que desde o primeiro dia em que decidi fazer mestrado sempre esteve ao meu lado, me apoiando, orientando. Foi um imenso prazer ter trabalhado ao seu lado. Muito Obrigada!

Ao projeto Via Digital, pela oportunidade de ser uma semente.

A família Anjos da Luz, que sempre estiveram na torcida mesmo a distância.

Aos meus colegas do FLOApp, Salatiel, Leonardo, Alan, Gabriel, Rafael, Bruno Coitinho e Anne Caroline.

Aos meus amigos Elvis, Patrício, Yuska e Marcos pelas conversas, amizade e carinho.

A Aninha e Vera pela atenção, presteza e dedicação com todos os alunos.

A todos os professores do DSC que sempre prestativos contribuíram para o meu desenvolvimento intelectual.

A CAPES pelo apoio financeiro.

A todos que contribuíram com a concretização deste trabalho. Muito Obrigada!

SUMÁRIO

LISTA DE ABREVIATURAS.....	VII
LISTA DE FIGURAS.....	VIII
LISTA DE QUADROS	IX
RESUMO	X
ABSTRACT	XI
CAPÍTULO 1 - INTRODUÇÃO.....	12
1.1 PROBLEMA.....	12
1.2 OBJETIVOS.....	14
1.3 ESTRUTURA DA DISSERTAÇÃO.....	14
CAPÍTULO 2 - FUNDAMENTAÇÃO TEÓRICA	16
2.1 SOFTWARE LIVRE.....	17
2.2 AMBIENTE DE DESENVOLVIMENTO COLABORATIVO.....	23
2.3 GESTÃO BASEADA EM PROCESSOS	32
2.4 CONSIDERAÇÕES FINAIS	33
CAPÍTULO 3 - O PROJETO VIA DIGITAL	35
3.1 INTRODUÇÃO	36
3.2 MODELO DE NEGÓCIO	38
3.3 MODELO DE NEGÓCIO VIA DIGITAL.....	42
3.4 CONSIDERAÇÕES FINAIS	43
<u>CAPÍTULO 4 - FLOAPP.....</u>	44
4.1 INTRODUÇÃO	45
4.2 ARQUITETURA DO FLOAPP	54
4.3 CARACTERIZAÇÃO DO FLOAPP.....	57
4.4 CONSIDERAÇÕES FINAIS	59
CAPÍTULO 5 - ESTUDO DE CASO	60
5.1 CONSIDERAÇÕES INICIAIS.....	61
5.2 EXECUÇÃO DO ESTUDO DE CASO	65
5.3 AVALIAÇÃO DOS RESULTADOS.....	68
5.4 CONSIDERAÇÕES FINAIS.....	75
CAPÍTULO 6 - CONCLUSÕES E TRABALHOS FUTUROS.....	78
6.1 RESUMO DAS CONTRIBUIÇÕES.....	79
6.2 TRABALHOS FUTUROS	80
REFERÊNCIAS BIBLIOGRÁFICAS.....	81
APÊNDICE: DETALHAMENTO DO MODELO DE NEGÓCIO EM PROCESSOS.....	87
ANEXO A. DELINEAMENTO DO PERFIL DO USUÁRIO – QUESTIONÁRIO USER (USER SKETCHER).....	99
ANEXO B. SONDAGEM SUBJETIVA DA SATISFAÇÃO DO USUÁRIO - QUESTIONÁRIO USE (USER SATISFACTION ENQUIRER)	100

LISTA DE ABREVIATURAS

ADC	AMBIENTE DE DESENVOLVIMENTO COLABORATIVO
BSD	BERKELEY SOFTWARE DISTRIBUTION
CN	COMUNIDADE DE NEGÓCIO
CRM	CUSTOMER RELATIONSHIP MANAGEMENT
CT	COMUNIDADE TÉCNICA
CVS	CONTROL VERSION SYSTEM
GC	GERÊNCIA DE CONFIGURAÇÃO
GPL	GNU GENERAL PUBLIC LICENSE
IES	INSTITUIÇÃO DE ENSINO SUPERIOR
LGPL	GNU LESSER GENERAL PUBLIC LICENSE
SL	SOFTWARE LIVRE
TI	TECNOLOGIA DE INFORMAÇÃO

LISTA DE FIGURAS

FIGURA 1 - A CATEGORIZAÇÃO DAS CARACTERÍSTICAS DE UM ADC.	25
FIGURA 2 - MODELO CONCEITUAL DE UM ADC (BOOCH,2002).....	26
FIGURA 3 - SUMÁRIO DAS INFORMAÇÕES DISPONÍVEIS PELO <i>SOURCE FORGE</i>	27
FIGURA 4 - ESTATÍSTICAS OFERECIDAS PELO <i>SOURCE FORGE</i>	28
FIGURA 5 - ETAPAS BÁSICAS DE UM PROCESSO (SENTANIN,2004).	32
FIGURA 6 - MODELO DE NEGÓCIO DISTRIBUIDOR (KRISHNA,2005).	40
FIGURA 7 - MODELO DE NEGÓCIO DESENVOLVEDOR DE SW (MODELO NÃO-GPL) (KRISHNA,2005).	40
FIGURA 8 - MODELO DE NEGÓCIO DESENVOLVEDOR DE SW (MODELO GPL) (KRISHNA,2005).	41
FIGURA 9 - MODELO DE NEGÓCIO FORNECEDOR DE SERVIÇOS (KRISHNA,2005).	42
FIGURA 10 - MODELO DE NEGÓCIO VIA DIGITAL.	43
FIGURA 28 – VISÃO GERAL DO FLOAPP.....	46
FIGURA 29 - RASTREAR COMPONENTES.	47
FIGURA 30 - VISÃO GERAL DO PROJETO.	48
FIGURA 31 - DOWNLOAD DO COMPONENTE.....	49
FIGURA 32 - CADASTRAR DEMANDA.	50
FIGURA 33 - VISÃO GERAL DE UMA DEMANDA.	52
FIGURA 34 - PROJETO ARQUITETURAL DO FLOAPP.	55
FIGURA 2 - FLUXO GERAL DO FLOAPP BASEADO NO PROCESSO "DESENVOLVIMENTO DE SL E GARANTIA DA QUALIDADE".	58
FIGURA 35 - MODELO DE INTERAÇÃO DO CONCURSO. (MUNIZ,2007).....	62
FIGURA 36 - DINÂMICA DO CONCURSO (MUNIZ,2007).	64
FIGURA 37 - AVALIAÇÃO GERAL DO USO DO FLOAPP.	71
FIGURA 38 - CADASTRO DE DEMANDAS.	71
FIGURA 39 - RASTREAMENTO DE COMPONENTES.	72
FIGURA 40 - CADASTRAR NOVO PROJETO.....	72
FIGURA 41 - DOWNLOAD DE COMPONENTES.	72
FIGURA 42 - SATISFAÇÃO AO EXECUTAR AS ATIVIDADES USANDO O FLOAPP.	73
FIGURA 43 - COMPREENSÃO DAS MENSAGENS DE ERRO EXIBIDAS PELO FLOAPP.	73
FIGURA 44 - SATISFAÇÃO COM O USO DO FLOAPP.	74
FIGURA 45 - FLUXOGRAMA DO PROCESSO PLANEJAMENTO.	88
FIGURA 46 - FLUXOGRAMA DO PROCESSO GERÊNCIA DO CONCURSO.	89
FIGURA 47 - FLUXOGRAMA DO PROCESSO GESTÃO E CAPTAÇÃO DE RECURSOS.	89
FIGURA 48 - FLUXOGRAMA DO PROCESSO CRM.	90
FIGURA 49 - FLUXOGRAMA DO PROCESSO PROSPECÇÃO.	91
FIGURA 50 - FLUXOGRAMA DO PROCESSO DISSEMINAÇÃO.	91
FIGURA 51 - FLUXOGRAMA DO PROCESSO VENDAS.....	92
FIGURA 52 - FLUXOGRAMA DO PROCESSO SERVIÇOS DE APOIO.	92
FIGURA 53 - FLUXOGRAMA DO PROCESSO GESTÃO DO DESEMPENHO.....	94
FIGURA 54 - FLUXOGRAMA DO PROCESSO GOVERNANÇA DO ECOSISTEMA.....	94
FIGURA 56 - FLUXOGRAMA DO PROCESSO ENGENHARIA DE REQUISITOS.....	95
FIGURA 57 - FLUXOGRAMA DO PROCESSO PROJETO DE COMPONENTES.....	96
FIGURA 58 - FLUXOGRAMA DO PROCESSO DESENVOLVIMENTO E TESTES.	97
FIGURA 59 - FLUXOGRAMA DO PROCESSO IMPLANTAÇÃO E CAPACITAÇÃO.	97
FIGURA 60 - FLUXOGRAMA DO PROCESSO PROJETO E CONSTRUÇÃO DE COMPONENTES DE GESTÃO.....	98

LISTA DE QUADROS

QUADRO 1 - TIPOS DE LICENÇAS E RESTRIÇÕES.....	18
QUADRO 2 - CARACTERÍSTICAS DOS ADCS E REPOSITÓRIOS EXISTENTES.....	31
QUADRO 21 - CONTRIBUIÇÕES DO FLOAPP.....	59
QUADRO 22 - INFORMAÇÕES BÁSICAS DA CIDADE DE FAGUNDES, PARAÍBA - BRASIL (IBGE,2006).....	62
QUADRO 23 - INFORMAÇÕES BÁSICAS DA CIDADE DE PATOS, PARAÍBA - BRASIL (IBGE,2006).....	62
QUADRO 24 - PARTICIPANTES DO CONCURSO PILOTO.....	65
QUADRO 25 - PARTICIPANTES DO CONCURSO REGIONAL.....	65
QUADRO 26 - DELINEAMENTO DO PERFIL DO USUÁRIO.....	70
QUADRO 27 - ANÁLISE DO INDICADOR DE SATISFAÇÃO SUBJETIVA DO USE. (OLIVEIRA,2004).....	74
QUADRO 28 - DIFERENÇAS ENTRE OS AMBIENTES EXISTENTES E O PROPOSTO.....	80
QUADRO 29 - DETALHAMENTO DO PROCESSO PLANEJAMENTO.....	87
QUADRO 30 - DETALHAMENTO DO PROCESSO GERÊNCIA DE CONCURSO.....	88
QUADRO 31 - DETALHAMENTO DO PROCESSO GESTÃO E CAPTAÇÃO DE RECURSOS.....	89
QUADRO 32 - DETALHAMENTO DO PROCESSO CRM.....	90
QUADRO 33 - DETALHAMENTO DO PROCESSO PROSPECÇÃO.....	90
QUADRO 34 - DETALHAMENTO DO PROCESSO DISSEMINAÇÃO.....	91
QUADRO 35 - DETALHAMENTO DO PROCESSO DE VENDAS.....	91
QUADRO 36 - DETALHAMENTO DO PROCESSO SERVIÇOS ADMINISTRATIVOS AUXILIARES.....	92
QUADRO 37 - DETALHAMENTO DO PROCESSO SERVIÇOS JURÍDICOS.....	92
QUADRO 38 - DETALHAMENTO DO PROCESSO GESTÃO DE RECURSOS FINANCEIROS.....	93
QUADRO 39 - DETALHAMENTO DO PROCESSO GESTÃO DO DESEMPENHO.....	93
QUADRO 40 - DETALHAMENTO DO PROCESSO GOVERNANÇA DO ECOSISTEMA.....	94
QUADRO 41 - DETALHAMENTO DO PROCESSO ENGENHARIA DE REQUISITOS.....	95
QUADRO 42 - DETALHAMENTO DO PROCESSO PROJETO DE COMPONENTES.....	96
QUADRO 43 - DETALHAMENTO DO PROCESSO DESENVOLVIMENTO E TESTES.....	96
QUADRO 44 - DETALHAMENTO DO PROCESSO IMPLANTAÇÃO E CAPACITAÇÃO.....	97
QUADRO 45 - DETALHAMENTO DO PROCESSO PROJETO E CONSTRUÇÃO DE COMPONENTES DE GESTÃO.....	98
QUADRO 46 - DETALHAMENTO DO PROCESSO SUPORTE TÉCNICO.....	98

RESUMO

O desenvolvimento de Software Livre se dá em ambientes colaborativos, no qual a comunidade técnica pode contribuir de forma descentralizada e integrada, tendo um maior controle e gerenciamento do que está sendo desenvolvido e distribuído para os usuários. Há, portanto, a necessidade de um ambiente diferenciado que possa atender melhor a comunidade técnica e a comunidade de negócio, proporcionando além das funcionalidades existentes, mais documentação e informações sobre a qualidade e reuso dos componentes. Neste trabalho é apresentado um modelo que visa inserir usuários ligados ao domínio de negócio no desenvolvimento de soluções livres. Para tanto é apresentado o ambiente de suporte FLOApp (Free/Livre/Open Applications).

Palavras-chaves: Software Livre, Modelo de Negócio e Ambiente Colaborativo.

ABSTRACT

Development of Free Software if made in cooperative environments, where technical community can contribute by decentralised and integrated fashion, allowing greater control and management of what is being developed and distributed for the users. Therefore, it has necessity of an common (or single) environment to support better technical and business communities, providing beyond the existing functionalities, more documentation and information on quality and reuse of the components. As main result of this work, presents a model aims to insert business users in free solutions development, available FLOApp (Free/Livre/Open Applications) as support environment.

Keywords: *Free Software, Business Model and Collaborative Environment.*

CAPÍTULO 1

INTRODUÇÃO

Um Software Livre é definido pela licença sob a qual ele é distribuído e os maiores esforços têm sido dirigidos para sistemas operacionais, ferramentas de desenvolvimento, componentes de rede e de infra-estrutura e softwares orientados para o ambiente *web*. Além dessas áreas, vêm surgindo novos Softwares Livres orientados para ferramentas para escritório e de produtividade (FELLER,2000).

As principais motivações tecnológicas para o Software Livre, conforme Golden (2005), são as necessidades de se aperfeiçoar os ciclos de desenvolvimento, de se aumentar a qualidade dos códigos gerados, tornando-os mais robustos, e de se buscar a adoção de padrões abertos. Além dessas, existe a motivação econômica que busca reduzir os custos através do desenvolvimento compartilhado.

Dentre as principais características encontradas nas soluções de Software Livre estão a rapidez de desenvolvimento devido a grande quantidade de participantes, a liberação incremental de novas versões e a constante evolução dos produtos. Por outro lado, os desenvolvedores envolvidos em determinado projeto, geralmente encontram-se dispersos geograficamente e, por isso, sua interação é realizada de forma virtual, através da Internet.

De acordo com Lima (2005), a metodologia adotada para a construção de Software Livre preconiza que o desenvolvimento e, principalmente, a depuração de código, sejam realizados de forma cooperativa e descentralizada. Isso é efetuado por um grande número de indivíduos, normalmente, de forma gratuita, utilizando como suporte ambientes *web*, os quais possibilitam o desenvolvimento colaborativo entre os participantes de uma comunidade de Software Livre.

1.1 PROBLEMA

Diante do cenário exposto por Lima e Golden, existem algumas demandas necessárias para a comunidade técnica e para a comunidade de usuários que ainda não são cobertas pelos ambientes colaborativos existentes.

O *Source Forge* (SFORGE,2005) é um dos maiores ambientes *web* de desenvolvimento colaborativos (ADC) do mundo. Seu principal diferencial está nas ferramentas de suporte disponíveis ao desenvolvedor pelo ADC. Entretanto, para os desenvolvedores e usuários finais, as estimativas e métricas fornecidas por estes ambientes são insuficientes. Em geral, são fornecidas apenas as estimativas abaixo para um determinado projeto:

- O número de *downloads* realizados;
- As falhas cadastradas;
- A frequência de entradas em um fórum da comunidade.

Para usuários finais, ou mesmo um novo desenvolvedor interessado em contribuir com a comunidade, que deseja obter informações sobre a qualidade e documentação de um projeto, as estimativas descritas acima são insuficientes para a tomada de decisão. Além destas, seria importante ressaltar, por exemplo, o grau de cobertura dos testes, quão reutilizável é o projeto e, sobretudo, indicadores de satisfação fornecidos por outros usuários, já que a satisfação dos desenvolvedores é a principal motivação para estes contribuir com as comunidades de Software Livre. As experiências também mostram que a satisfação dos usuários está diretamente ligada ao sucesso do projeto.

Fazendo uma analogia à forma como é distribuído (comercializado) o software proprietário, tem-se que estes são colocados em “prateleiras” onde todas as informações do software são transparentes aos consumidores (e mesmo assim alguns usuários ainda sentem-se confusos quanto ao produto). Portanto, como os usuários podem tomar qualquer decisão em um ADC, se as informações que estão disponíveis não são objetivas e claras?

Através do Projeto Via Digital (apresentado no capítulo 3), foi proposto um modelo de gestão de negócio que promove a auto-organização dos atores em duas comunidades (fomentando a interseção entre elas): comunidade técnica (formada pelos desenvolvedores) e a comunidade de negócio (formada pelos usuários finais).

Nesse contexto, o propósito deste trabalho é propor um ambiente orientado ao modelo de gestão proposto, fomentando e facilitando a colaboração das comunidades envolvidas na produção de componentes para o domínio do negócio.

1.2 OBJETIVOS

O objetivo deste trabalho é viabilizar a participação da comunidade de negócio no processo de desenvolvimento de componentes livres, suportando a interação e colaboração com a comunidade técnica.

No sentido de atingir o objetivo principal vamos apresentar um ambiente baseado em uma arquitetura orientada a *plugins*, com a qual será possível adicionar novas funcionalidades (ferramentas) de acordo com as necessidades das comunidades.

O ambiente possui os seguintes objetivos específicos:

- ❖ Permitir o rastreamento de falhas
- ❖ Permitir a integração de código
- ❖ Permitir a criação e execução de testes
- ❖ Fornecer métricas de documentação
- ❖ Fornecer métricas de reuso
- ❖ Permitir as comunidades técnicas e de negócio a qualificação de demandas e componentes respectivamente
- ❖ Permitir a gestão de mudanças
- ❖ Privacidade e Segurança
- ❖ Permitir o rastreamento de componentes
- ❖ Permitir a gestão de demandas
- ❖ Fornecer indicadores de satisfação
- ❖ Fornecer uma FAQ (Dúvidas mais frequentes)

1.3 ESTRUTURA DA DISSERTAÇÃO

Esta dissertação está organizada em seis capítulos. O capítulo 1 traz a introdução do trabalho e define o nosso escopo de pesquisa.

O Capítulo 2 traz a fundamentação teórica da pesquisa que apresenta os diversos conceitos que embasam todo o trabalho. Nesse capítulo são discutidos os assuntos referentes à Software Livre, o que é, quem desenvolve, qual a metodologia de desenvolvimento adotada e onde são armazenados. Em seguida são abordados os conceitos dos ambientes de desenvolvimento colaborativo, destacando as características dos existentes. Por fim, são apresentados os principais conceitos envolvendo a gestão baseada em processos.

O Capítulo 3 apresenta o projeto Via Digital como contexto em que este trabalho esteve inserido. É destacada a visão de negócio envolvendo o desenvolvimento de Software Livre e com um maior detalhamento é apresentado o modelo de negócio Via Digital.

No Capítulo 4 é apresentado o ambiente que está sendo proposto, suas funcionalidades, os atores envolvidos e por fim o detalhamento da arquitetura.

O Capítulo 5 traz o estudo de caso realizado em duas edições do concurso de Software Livre promovido pelo projeto Via Digital, utilizando o ambiente proposto apenas na segunda edição.

No Capítulo 6 são apresentadas as conclusões e são discutidos os trabalhos futuros.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos relevantes que foram utilizados como base para realização do trabalho de pesquisa pertinente a esta dissertação de mestrado. Assim, é colocado como fundamentos teóricos deste trabalho; conceitos, terminologias e caracterizações de: Software Livre, Ambiente de Desenvolvimento Colaborativo e Gestão Baseada em Processos. Nesta etapa buscamos:

- ❖ Apresentar os conceitos de Software Livre, o perfil dos seus desenvolvedores, qual (ou quais) a (as) metodologia(s) de desenvolvimento adotada (s) pelas comunidades, os ambientes de suporte ao desenvolvimento utilizados.
- ❖ Argumentar sobre os conceitos pertinentes a ambientes colaborativos:
 - Mostrar características dos ambientes existentes.
- ❖ Caracterizar a importância da gestão por processos para o ambiente colaborativo.

2.1 SOFTWARE LIVRE

O desenvolvimento de Software Livre (SL), a partir das décadas de 60 e 70, começou a ser realizado de forma colaborativa e aberta em diversas instituições e empresas. O Unix BSD (2007) é um exemplo desta tendência (MCKUSICK,1999). Embora as licenças não explicitassem claramente liberdade, a redistribuição do software era vista como positiva e o software geralmente era fornecido com o seu código fonte Raymond (1999) e Stallman (1999). Segundo a *Free Software Foundation* (FSWF,2005), Software Livre é um software que é distribuído acompanhado de código fonte e que pode ser livremente modificado e redistribuído.

Software Livre utiliza licenças e copyright com o objetivo, segundo Lerner (2005), de garantir a liberdade do software a seus usuários. Cada licença possui detalhes diferentes, a Quadro 1 apresenta as principais diferenças entre as licenças mais utilizadas.

Licença	Restrições
Domínio Público	Efetivamente qualquer software sem licença é de domínio público, que permite que seu usuário faça o que queira com ele, inclusive modificar e redistribuir. Não há nenhum tipo de restrição sobre o software, e pode ser modificado e distribuído sem código fonte, por exemplo.
BSD (2007) - <i>Berkeley Software Distribution</i>	Permite redistribuição livre do software. Ocasionalmente inclui uma cláusula que obriga cópias redistribuídas a manterem um aviso do copyright. Não obriga versões modificadas a serem livres, e nem a fornecerem código fonte.
GPL (2007) - <i>GNU General Public Licence</i>	Permite redistribuição desde que mantida a garantia de liberdade inalterada aos usuários da cópia redistribuída. Obriga versões modificadas a serem livres, e portanto, a serem fornecidas acompanhadas de código fonte.

Licenças	Restrições
LGPL (2007) - <i>GNU Lesser General Public License</i> e MPL (2007) - <i>Mozilla Public Licence</i>	Permite redistribuição do código em si mantida a garantia de liberdade inalterada. No entanto, permitem que este código seja usado em um “produto maior” sem que este tenha que ser licenciado livremente. Se modificações forem feitas ao código em si, devem ser fornecidas acompanhadas de código fonte. Esta restrição não cobre o código fonte do “produto maior”.

Quadro 1 - Tipos de Licenças e Restrições.

Um dos aspectos interessantes de Software Livre é que em nada restringe o preço do software. Isto significa que é perfeitamente possível cobrar um valor pelo software ou por sua distribuição. Na prática, o fato da redistribuição do software ser irrestrita resulta em um TCO (custo total de posse) suficiente para motivar as pessoas a não copiarem o software de alguém que já o tem. Em geral, o que se observa é que existem duas formas principais de se adquirir o software:

- ❖ Transferindo o software de um repositório *on-line* na Internet.
- ❖ Adquirindo o software de um distribuidor, que cobra uma taxa pela distribuição. Em muitos casos o software é oferecido em uma embalagem e acompanhado de manuais e mídia digital.

Podemos definir ainda o termo código aberto (*Open Source*), o qual descreve melhor a forma de desenvolvimento que é aplicada em grande parte dos projetos de Software Livre. Em ambos os conceitos, Software Livre e Código Aberto, as mesmas premissas são mantidas a liberdade de copiar, distribuir, modificar e melhorar o software. Diferentes interpretações podem ser dadas a Software Livre dependendo do contexto em questão (FSWF,2005). Apoiado nesta afirmação, como a finalidade deste trabalho está voltada para aspectos do desenvolvimento de projetos de SL, ao realizarmos referências a Software Livre também estaremos referenciando código aberto.

Diante destes conceitos acima, podemos juntamente com as definições utilizadas pelo *Source Forge* (SFORGE,2007) e *Freash Meat* (FRESMEAT,2007) definir um projeto de Software Livre como sendo o código fonte, que é o software propriamente dito, qual possui inúmeras cópias entre todos seus usuários e repositórios de dados, e

uma comunidade de desenvolvedores, que trabalham para codificar e corrigir este software. Estes desenvolvedores em conjunto, colaboram através da Internet. Os desenvolvedores podem ter status diferenciado dentro do projeto (FIELDING,1999). Tendo como cliente e futuros desenvolvedores os usuários do software. Apesar de todo software ter usuários, a participação destes no Projeto de Software Livre é diferenciada e essencial; os usuários discutem inovações e reportam os erros encontrados do projeto. Através deste mecanismo, os desenvolvedores são motivados a corrigir os erros, desenvolvendo portanto um produto melhor (RAYMOND,1999). Em ambientes colaborativos, nos quais desenvolvedores concentram todas as informações referentes ao projeto, assim como acompanhamento das *releases* anteriores, algumas estatísticas, listas de discussões e falhas, os usuários podem procurar por algum projeto do seu interesse e ter acesso às informações acima, como também ao próprio código da aplicação

2.1.1 DESENVOLVIMENTO DE SOFTWARE LIVRE

Raymond (RAYMOND,1999) em (*The Cathedral and the Bazaar*) descreve dois modelos aparentemente disjuntos de desenvolvimento de software. O lado catedral é o desenvolvimento de software tradicional: centralizado, burocrático, fechado, processo pesado, etc. Do lado oposto, o modelo bazar é software livre: distribuído, baseado em meritocracia, esforço em troca de reputação, em muitos casos nada burocrático e um processo totalmente aberto e mais leve.

Segundo Raymond (1999), não é adequado codificar desde o início no estilo bazar. A comunidade precisa de algo executável e já com características próprias para testar e utilizar. Para se construir uma comunidade, é necessário apresentar uma promessa plausível, convencer os co-desenvolvedores de que o projeto tem potencial para fazer aquilo que ele pretende. Mais ainda, o desenvolvedor que utiliza apenas sua capacidade mental para um projeto fechado irá ficar atrás dos desenvolvedores que saibam criar um contexto aberto e evolutivo, no qual a visualização de erros e melhorias sejam feitas por centenas de pessoas.

Raymond (1999) acredita, que o futuro do Software Livre irá pertencer a pessoas que saibam se comunicar e criar comunidades para seus projetos. Nenhum desenvolvedor de código fechado pode competir com um conjunto de talentos reunidos. A cultura de software livre deve triunfar, não porque a cooperação seja moralmente

correta ou a proteção do software seja moralmente errada, mas simplesmente porque o mundo do software fechado não pode vencer uma corrida evolucionária contra comunidades de software aberto, cuja forma de desenvolvimento aloca tantos recursos e tempo hábil, que conduz os problemas a um patamar em que é muito mais fácil resolvê-los.

Ainda observando o modelo de desenvolvimento de comunidades de SL, percebemos que não existe uma metodologia de desenvolvimento universal para SL. Em Lima (2005) foram organizadas um conjunto de práticas nas áreas de coordenação, gerência de configuração, gerência da evolução de software e lançamento de versões. As práticas foram analisadas segundo a escolha de algumas comunidades de Software Livre proeminentes, refletindo diretrizes de trabalho que projetos utilizam em seu ciclo de vida.

Coordenação

A coordenação de projetos de software diz respeito ao gerenciamento de recursos em uma organização. Quem coordena os projetos deve solucionar problemas técnicos e não-técnicos, utilizando os recursos que possui de maneira eficaz. A atividade gerencial exige que seu executor motive pessoas, planeje e organize o trabalho de forma que o mesmo seja feito adequadamente (SOMMER,2003).

Gerência de Configuração

A Gerência de Configuração (GC) permite que várias pessoas trabalhem juntas e em um mesmo projeto de maneira mais eficiente. A GC deve permitir que desenvolvedores trabalhando juntos compartilhem o mesmo código - o esforço de desenvolvimento de um módulo é compartilhado, que se tenha acesso às versões mais estáveis do sistema, e que também seja possível voltar à versões anteriores - permitindo que as mudanças realizadas em um módulo ou componente sejam reversíveis para o estado anterior a estas mudanças.

Gerência de Evolução

A gerência de evolução em projetos de Software Livre, de acordo com Lima (2005), é suportada por sistemas de rastreamento e acompanhamento de mudanças (*tracking systems*). Estes sistemas são instrumentos utilizados por comunidades de desenvolvimento de Software Livre para reportar falhas no software em relação as falhas de funcionamento, notificações de correções efetuadas e requisições de funcionalidade, consistindo em funções importantíssimas para o processo de evolução do software. Qualquer membro da comunidade pode relatar uma falha e registra-la em um sistema de rastreamento e acompanhamento de mudanças, bastando apenas criar uma conta de acesso à ferramenta para isso. Desta maneira, todos colaboram no sentido de manter o software com um funcionamento estável. Na medida em que falhas são cadastradas na base de dados do sistema passa por um processo de triagem, que é feito por desenvolvedores mais maduros dentro da comunidade.

Lançamento de versões

O lançamento de versões, segundo Lima (2005), consiste em disponibilizar versões, estáveis ou não, do software produzido por uma determinada comunidade. Para tal, o software encontra-se em formatos de distribuição que ofereçam maior facilidade de acesso, *download*, instalação e configuração para seus usuários. Esta prática é realizada durante a vida de uma comunidade em atividade.

2.1.2 QUEM SÃO OS DESENVOLVEDORES DE SOFTWARE LIVRE?

Várias pessoas acreditam que os desenvolvedores de SL são apenas estudantes ou pessoas fora do mercado, com a suposição de que eles trabalham em SL ao invés de um trabalho real. Um estudo realizado pelo *Boston Consulting Group* (BOSTON, 2002) mostrou que desenvolvedores de Software Livre são motivados pela curiosidade intelectual e pelo desejo de melhorar suas habilidades. A caracterização dos desenvolvedores se reflete entre uma maioria de homens entre os vinte e trinta anos de idade, com uma média de onze anos de experiência como profissionais em TI. Em termos de localização geográfica, o estudo mostrou ainda que a América do Norte concentra 46% dos desenvolvedores de código aberto, a Europa com 42%, e os 12% restantes estão localizados em outros países.

Segundo Golden (2005), a maioria dos desenvolvedores é constituída de profissionais que trabalham em organizações de tecnologia. No entanto também existem desenvolvedores de Software Livre que trabalham em organizações que usam produtos de código aberto e estes se tornam membros de alguma comunidade, de forma a melhorarem o produto utilizado pela empresa a qual fazem parte. Os estudantes também são desenvolvedores. Portanto, grande parte dos desenvolvedores já trabalha com tecnologia, e dedica um tempo significativo aos esforços de Software Livre.

Esses desenvolvedores, geralmente fazem parte de alguma comunidade ligada ao projeto de interesse. O conceito de comunidade é um dos aspectos mais importantes relacionados a Software Livre. Em Golden (2005), temos que uma comunidade é uma organização livre, formada por todos que estejam interessados em um produto em particular por alguma razão, não existindo nenhum requisito formal para participar.

Entretanto, como afirma Golden (2005), a falta de formalização não significa que não existam normas, padrões de colaboração. De um membro da comunidade espera-se interagir respeitosamente, acima de tudo, para contribuir como também aproveitar as vantagens oferecidas pela comunidade.

Podemos então classificar uma comunidade de Software Livre em dois grupos, o primeiro compreenderia a comunidade técnica, qual é formada pelos desenvolvedores, comunidades, empresas de TI e pessoas com algum conhecimento tecnológico referente a um determinado projeto; o segundo grupo seria a comunidade de negócios, formada por usuários finais de projetos que nem sempre possuem um vasto conhecimento em tecnologias. Há a necessidade que a comunidade de desenvolvedores (e ambientes nos quais os projetos se encontram disponíveis) lhes forneça algumas informações para orientá-los em termos de documentação, instalação, configuração e algumas métricas úteis, de forma a lhe dar subsídios sobre qual solução se adequa às suas necessidades e como proceder para adquiri-la.

2.1.3 ONDE RECUPERAR SOFTWARE LIVRE?

Software Livre é disponibilizado em diversos lugares diferentes. Um projeto SL pode ter seu próprio sítio para disponibilizar o produto. Existem vários portais que atuam como repositórios de Software Livre. E vários produtos de SL são disponibilizados

nesses portais, tornando conveniente a localização do produto através da ferramenta de busca do portal.

“O lugar mais conveniente para recuperar um produto é a partir de um portal de Software Livre” (GOLDEN,2005).

- ❖ **Sítios individuais de produtos SL:** Alguns produtos de Software Livre muito bem estabelecidos têm seus próprios sítios, que atuam como o principal mecanismo de distribuição do software, além de concentrar todas as interações entre as comunidades de negócios e técnica. Os Web Sites geralmente possuem fóruns para discussões e questões entre a comunidade. Assim como também são disponibilizadas notícias sobre o produto.

- ❖ **Portais de Software Livre:** Portais de SL oferecem uma localização centralizada para os produtos de Software Livre, oferecendo vários serviços e assim tornando o desenvolvimento e manutenção dos projetos mais simples. Golden (2005) afirma que o fato de ter vários projetos hospedados em um único portal oferece um valor real para os usuários - eles podem facilmente buscar e filtrar soluções dentre centenas e milhares de projetos.

2.2 AMBIENTE DE DESENVOLVIMENTO COLABORATIVO

Existem vários lugares para disponibilizar o Software Livre. Ele pode estar disponível em um site próprio do produto, ou estar disponível em algum ambiente colaborativo. Estes ambientes, tais como *Source Forge* (SFORGE 05), oferecem diversos serviços tornando tanto o início como a manutenção de um projeto de código aberto mais simples.

Ambiente de desenvolvimento colaborativo, de acordo com Margerison (2005), é geralmente um conjunto de ferramentas *web* que podem ser integradas dentro de um pacote coerente. Seu objetivo principal é prover um ambiente eficiente no qual equipes possam colaborar a fim de disponibilizar projetos de software.

De acordo com Booch (2002), um ambiente de desenvolvimento colaborativo é um espaço virtual onde todos os *stakeholders*¹ de um projeto - mesmo se distribuídos pelo tempo ou pela distância - podem negociar, realizar *brainstorms*, discutir, compartilhar o conhecimento, geralmente trabalhar junto para realização de alguma tarefa, mais frequentemente entregar algum executável e seus artefatos de suporte.

O valor agregado a um ambiente de desenvolvimento colaborativo está diretamente ligado ao potencial de manter todas as informações e recursos do projeto, gerenciar ferramentas e artefatos responsáveis por gerar os recursos e permitir a comunicação entre a comunidade de desenvolvimento, como observado em Booch (2002).

Mais especificadamente, temos os seguintes elementos essenciais para a construção de um ADC, sugerido por Fournier (2001):

- ❖ Ferramentas de mensagens instantâneas
- ❖ Fóruns de discussões e listas de *e-mails*
- ❖ Gerenciamento das informações centralizadas
- ❖ Mecanismo de busca e indexação dos artefatos
- ❖ Controle de mudanças dos artefatos
- ❖ Cronograma
- ❖ Notificação de eventos *on-line*
- ❖ Traçar o perfil dos recursos do projeto
- ❖ Qualificação
- ❖ Vários níveis de visualizar a informação
- ❖ Métricas e o panorama do projeto
- ❖ Auto-publicação de conteúdo
- ❖ Conteúdo personalizado

Em Booch (2002), estes elementos podem ser organizadas em três categorias. Estas são baseadas na natureza de coordenação, colaboração e comunicação de um ADC (Figura 1).

¹ Designa todas as pessoas ou empresas que, de alguma maneira, são influenciadas e tem interesse nas ações de uma organização. (ROZANSKI,2005)

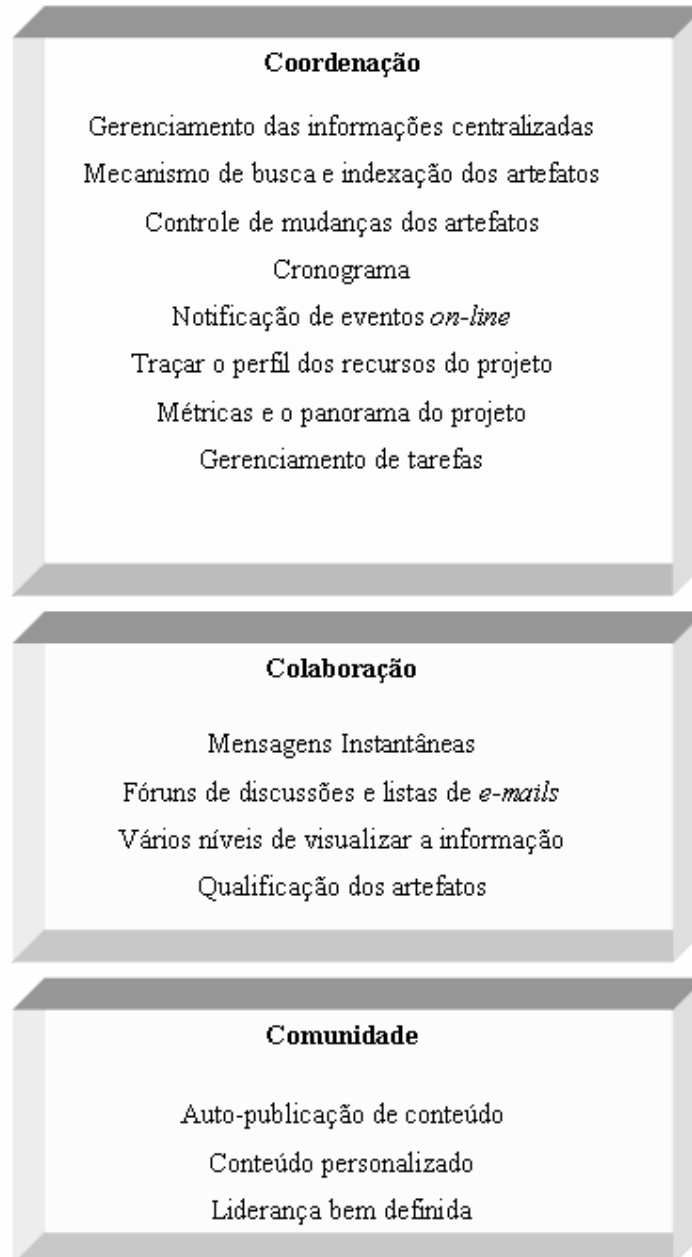


Figura 1 - A categorização das características de um ADC.

Na Figura 1 temos um conjunto de características agrupadas por camadas, as quais oferecem desafios significativos na integração técnica destas, assim como favorecem que um usuário experiente esteja satisfeito com as comunidades de desenvolvedores, uma vez que empregam rotineiramente muitas destas características para terminar uma tarefa compartilhada.

Conceitualmente, Booch (2002) apresenta um modelo conceitual de um ADC com foco em alguns aspectos: assegurar a integridade do software, dos artefatos e dos usuários validados (os quais possuem perfis diferentes, de forma que apenas alguns artefatos podem ser acessados). Um ADC é decomposto em três camadas:

- ❖ *Workspace* do projeto: provê mecanismos de gerência e colaboração.
- ❖ Ferramentas: provê uma infra-estrutura de suporte para gerência de mudanças, controle de versão e integração.
- ❖ Recursos de desenvolvimento: provê ferramentas para artefatos específicos ao desenvolvimento.

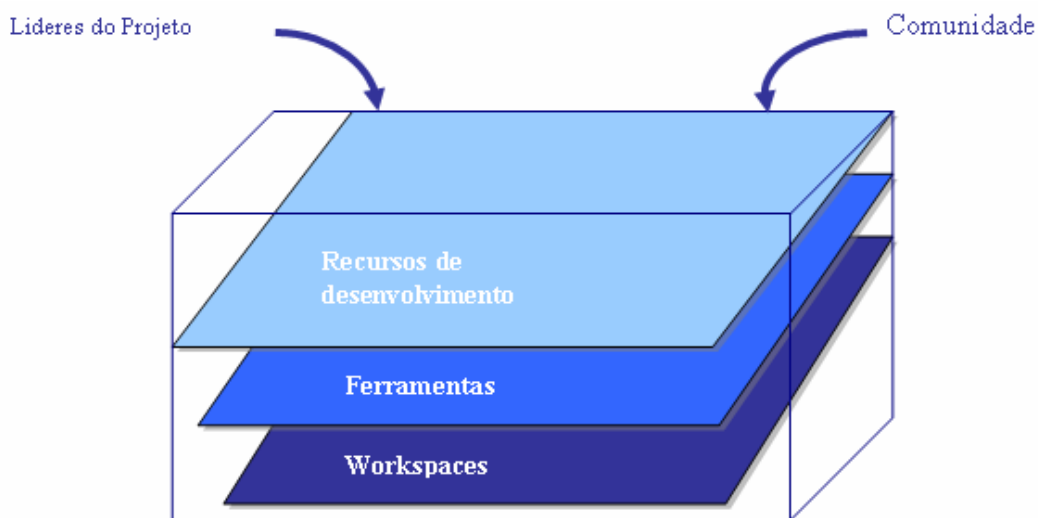


Figura 2 - Modelo Conceitual de um ADC (BOOCH,2002).

2.2.1 EXEMPLO DE UM AMBIENTE COLABORATIVO *SOURCE FORGE*

O Source Forge (SFORGE,2007) é um dos maiores ambiente *web* de desenvolvimento colaborativo do mundo. Desde 2001, a VA Linux (VALINUX,2007) restringiu o acesso ao código, tornando-o seu desenvolvimento privado sob a licença *SourceForge Enterprise Edition*. Ele faz parte da tecnologia OSDN (*Open Source Developer Network*), dando suporte *web* ao desenvolvimento de Software Livre e aos profissionais de TI. O *Source Forge* atualmente possui mais de 1 milhão de usuários registrados e 100 mil projetos de Software Livre em desenvolvimento.

Com a restrição ao código um novo projeto foi criado, o GForge (GFORGE,2007) , contendo as mesmas funcionalidades oferecidas pelo Source Forge antes de ter seu código fonte fechado. Conseqüentemente, o GForge é utilizado por várias empresas, projetos de software como seu repositório e o Source Forge utilizado por todos interessados em divulgar seus projetos e idéias.

No Source Forge, a comunidade de desenvolvedores pode encontrar as seguintes funcionalidades:

- ❖ Habilidade para organizar múltiplos projetos e equipes dentro de um repositório central.
- ❖ Gerência de controle de mudança (um repositório *web* com acesso remoto, predominantemente é usado o CVS, integrado a outras ferramentas do ambiente).
- ❖ Notas de versões.
- ❖ Requisições de falhas e melhorias.
- ❖ Ajuda e suporte.
- ❖ Listas de atividades.
- ❖ Estatísticas de *download*.

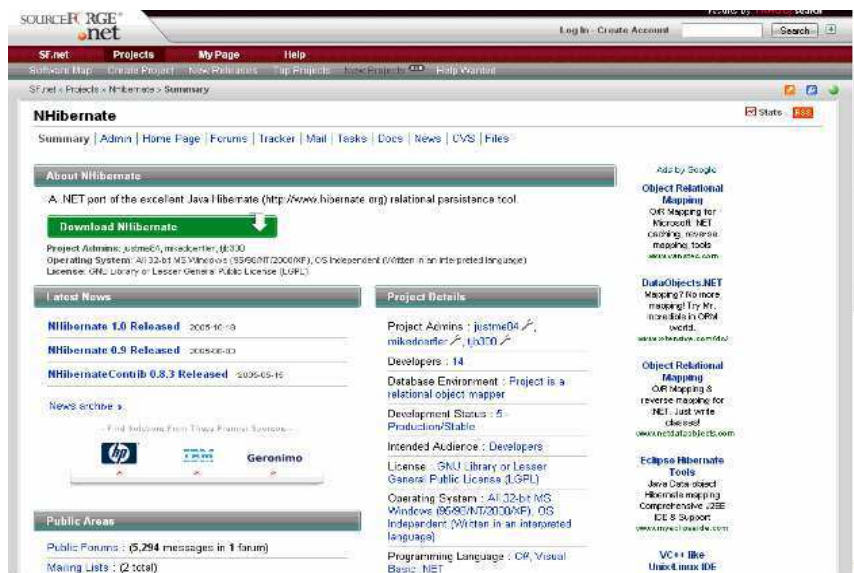


Figura 3 - Sumário das informações disponíveis pelo *Source Forge*.

A Figura 3 mostra o sumário das informações disponíveis para as comunidades, relativas aos projetos cadastrados. As informações são as seguintes:

- ❖ Status do projeto
- ❖ Sistema Operacional
- ❖ Linguagem de Programação
- ❖ Informações sobre os desenvolvedores
- ❖ Últimas notícias
- ❖ Lista de Falhas
- ❖ Requisições de melhorias
- ❖ Fórum e listas de emails

O *Source Forge* também disponibiliza algumas estatísticas, de modo que as comunidades possam ter uma visão geral dos elementos chaves dos projetos.

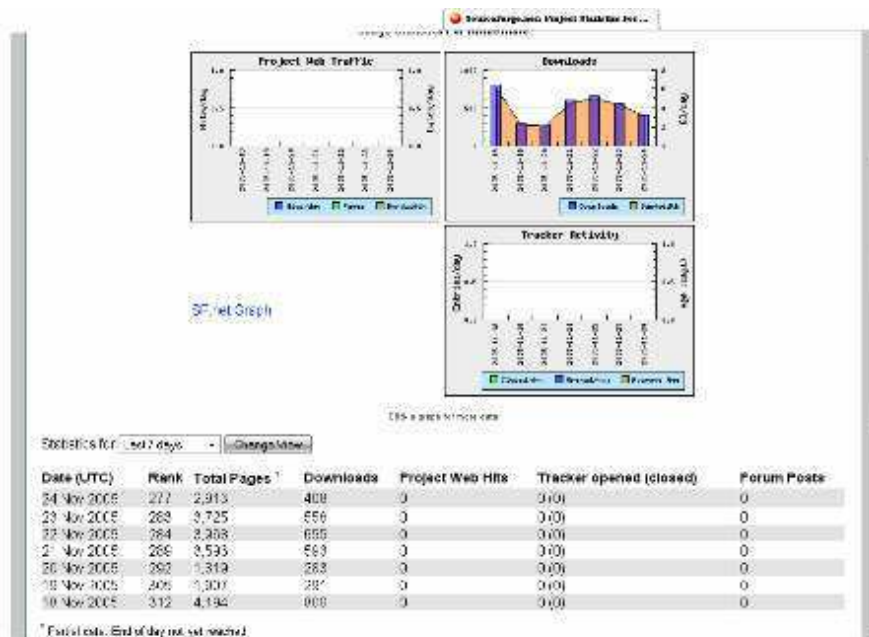


Figura 4 - Estatísticas oferecidas pelo *Source Forge*.

Na Figura 4 as comunidades podem observar quão ativo é um projeto em relação às diversas visões e *downloads*. Por exemplo, caso seja entregue a comunidade um versão do projeto, pode-se monitorar quantos *downloads* foram realizados sobre aquela versão. Caso possua poucos *downloads*, rapidamente à comunidade técnica poderá recuperar o envolvimento com outras comunidades e receber avaliações do que está disponibilizando.

Em síntese, o Quadro 2 apresenta as principais características, descritas abaixo, presentes no *Source Forge* em comparação com outros repositórios existentes.

- ❖ Rastreamento de Falhas: Os usuários podem reportar os defeitos encontrados no componente.
- ❖ Geração de código: Plataforma para desenvolvimento (*IDE - Integrated Development Environment*).
- ❖ Pacote de Distribuição: Gera um release da versão presente no repositório de forma automática e esta é disponibilizada na página do projeto para *download*.
- ❖ Integração de código: Integra de forma automática os módulos do componente gerando um sistema único integrado.
- ❖ Teste (execução e cobertura): Permite a criação dos casos de testes e execução de forma automática. Após a execução dos testes o resultado pode ser visualizado através de um relatório mostrando os sucessos, falhas e o grau de cobertura dos testes em relação ao código existente.
- ❖ Métricas de documentação: Os usuários podem através desta ferramenta identificar qual componente possui documentação (manuais de instalação e configuração, guia para o usuário, diagramas de modelagem e a própria documentação do código). E ainda analisar através das estatísticas fornecidas qual documentação apresenta suficiente para o uso do componente.
- ❖ Métricas de reuso: Fornece informações de forma a ajudar um usuário sobre o grau de reuso de um componente. Para tanto, são observados o uso de padrões de projeto, componentes, e a modularização da arquitetura.
- ❖ Qualificação: Após a realização do *download* de um componente, os usuários podem qualificá-lo de acordo com documentação, instalação/configuração e uso, através da atribuição de notas de zero (péssimo) a dez (Ótimo).
- ❖ Controle de versão: Ao colocar um componente no repositório é criado um histórico de todas as mudanças realizadas, permitindo a recuperação de qualquer versão.
- ❖ Privacidade: Cada usuário possui sua página com todas as tarefas e informações relacionadas ao seu perfil. Esta página possui acesso restrito ao proprietário.

- ❖ **Segurança:** Todas as informações contidas no ambiente são agrupadas através de perfis e de acordo com os privilégios de cada usuário, este poderá cadastrar um projeto, disponibilizar um release, inserir uma release no repositório entre outras.
- ❖ **Rastreamento de componentes:** Realiza buscas de componentes através de palavras chaves e/ou categorias entre os repositórios cadastrados.
- ❖ **Gestão de demandas:** Cadastra, atualiza, remove e lista as necessidades da comunidade de negócio.
- ❖ **Indicadores de satisfação:** Os usuários atribuem uma nota de zero (totalmente insatisfeito) a cinco (totalmente satisfeito) indicando seu nível de satisfação em relação ao componente.

Ambos *GForge* (GFORGE,2007) e *FreshMeat* (FRESHMEAT,2007) possuem características semelhantes ao *Source Forge*, no entanto eles apenas utilizam um subconjunto das ferramentas oferecidas pelo (SFORGE,2007). O *FreshMeat* destaca-se por oferecer à comunidade suporte a indicadores de qualificação e vários ranking baseados em métricas de atividade e *download* de produtos. O Código Livre (CODLIVRE,2007) é um repositório de projetos de Software Livre do Brasil, apresentando apenas ferramentas de gestão de banco de dados e busca. Por fim, o Collab.Net (COLLAB,2007) qual dispõe de ferramentas apenas para a gestão de mudanças.

	<i>Source Forge</i>	<i>GForge</i>	<i>FreshMeat</i>	Código Livre	Collab.Net
Comunidade Técnica					
Rastreamento de falhas	x	x	x		
Geração de código	x	x	x		
Pacote de distribuição	x	x	x	x	x
Integração de código					
Teste (execução e cobertura)					
Métricas de documentação					
Métricas de reuso					

	<i>Source Forge</i>	<i>GForge</i>	<i>FreshMeat</i>	Código Livre	Collab.Net
Comunidade Técnica					
Qualificação			X		
Controle de versão	X	X	X	X	X
Privacidade	X	X	X	X	X
Segurança	X	X	X	X	X
Comunidade de Negócio					
Rastreamento de componentes	X	X	X	X	X
Gestão de demandas					
Indicadores de satisfação					
FAQ (Dúvidas Frequentes)					

Quadro 2 - Características dos ADCs e repositórios existentes.

Ambos *Source Forge* e *GForge* se destacam por disponibilizar um conjunto de ferramentas de apoio a gestão de projetos de Software Livre. Nestes ambientes não existe a preocupação com a qualidade do software que é disponibilizado e nem com o usuário final do mesmo. Tanto as ferramentas e indicadores fornecidos são utilizados por usuários com vasto conhecimento em TI. Visto que os ambientes acima citados não apresentam nenhum manual ou suporte a estas ferramentas. Todo o material de suporte oferecido é de respeito ao próprio software que está sendo desenvolvido, quando de fato existe.

O *FreshMeat* diferentemente de *Source Forge* e *GForge* disponibiliza indicadores de qualificação, que são coletados com a ajuda da própria comunidade. Entretanto, não é suficiente para garantir a qualidade do produto. Esse ambiente também apresenta as mesmas deficiências relativo aos usuários finais.

O Código Livre (2007) é apenas um repositório, dispondo apenas de um banco de dados ordenado para a comunidade de usuários. Esta pode submeter o código fonte para o repositório, assim toda a comunidade poderá acessar, baixar e contribuir. No entanto, o Collab.Net (2007) fornece além do repositório ferramentas para gestão de configuração.

Logo, observamos que os ambientes de maior visibilidade na comunidade de Software Livre, não fornecem uma visão de negócio, uma vez que o cliente é a própria

comunidade técnica. As ferramentas disponíveis não garantem a qualificação do produto e tão pouco asseguram sua usabilidade por parte dos usuários finais.

2.3 GESTÃO BASEADA EM PROCESSOS

De acordo com Cruz (2003), processo é um conjunto de atividades que tem por objetivo transformar insumos (entradas), adicionando-lhes valor por meio de procedimentos, em bens ou serviços (saídas) que serão entregues e devem atender aos clientes.

A figura 5 apresenta um modelo básico das etapas de um processo.

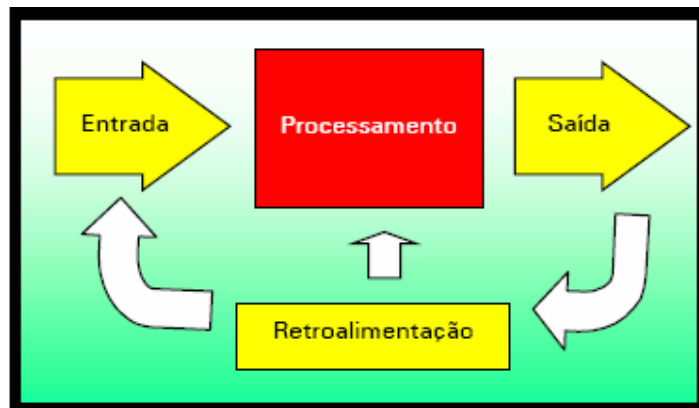


Figura 5 - Etapas básicas de um processo (SENTANIN,2004).

Verifica-se portanto, que para organizar um ambiente por processos, é necessário colocar o foco no cliente, já que os processos nele começam e terminam.

Hammer (HAMMER,1998) destaca a importância dos clientes para organizar um ambiente por processos uma vez que, para eles os processos constituem a essência do ambiente. O cliente não está interessado na estrutura organizacional, mas nos produtos e serviços produzidos por seus processos.

Isso demonstra a importância dos processos para uma organização, ambiente, pois eles permitem enxergar uma linha de atividades que começa com o entendimento exato do que o cliente deseja e que termina com o cliente adquirindo aquilo que precisa e deseja de um negócio (GONÇALVES,2000).

A gestão do negócio a partir de uma abordagem por processos procura mapear, analisar, melhorar e automatizar os processos para criar uma gestão integrada e eficaz

do negócio, medindo os resultados através de indicadores de desempenho e metas alinhados à estratégia corporativa (SENTANIN,2004).

A identificação e a análise de todos os processos, segundo Souza (2000) são fundamentais para o desenvolvimento de um sistema de gestão organizacional voltado para o alto desempenho.

Ainda em Souza (2000) observamos que a análise de processos leva ao melhor entendimento do funcionamento da organização e permite a definição adequada de responsabilidades, a utilização eficiente dos recursos, a prevenção e solução de problemas, a eliminação de atividades redundantes e a identificação clara dos clientes e fornecedores. A constante análise do sistema de gestão, dos processos e da interdependência entre eles ajuda a desenvolver o pensamento sistêmico na organização.

A base para a tomada de decisão, em todos os níveis da organização, é a análise de fatos e dados gerados em cada um de seus processos. Estes se transformam em informações relacionadas a todos os aspectos importantes para a organização, ou seja, clientes, mercados, produtos, processos, fornecedores, sociedade e ou aspectos financeiros.

O conhecimento adquirido por meio das informações é retido pela organização para que possa funcionar de maneira mais ágil e independente (SOUZA,2000).

Um sistema estruturado de informação desenvolve métodos de comparação de práticas e de análise crítica, introduz inovações ou melhorias de forma mais rápida e toma decisões mais eficazes.

2.4 CONSIDERAÇÕES FINAIS

Neste capítulo foram discutidos conceitos proeminentes do Software Livre, desde os diferentes conceitos de SL, como as licenças podem intervir no seu uso. Ainda foi abordado um conjunto de práticas adotadas pelas comunidades visto que não existe um processo formal seguido por todas. Foi identificado o perfil dos desenvolvedores de Software Livre, caracterizado como estes se organizam criando o conceito de comunidade. A partir desse conceito, foram caracterizadas dois tipos de comunidades, a de negócio, fornecendo uma visão para os usuários finais leigos em TI e a técnica, voltada para as comunidades de desenvolvedores existentes.

Após a caracterização das comunidades de Software Livre, foi apresentada como estas se organizam para disponibilizar o produto (os *releases* do projeto) objetivando a maior participação da comunidade.

Desta forma foi realizada uma análise sobre os ambientes colaborativos e repositórios utilizados pelas comunidades, abordando inicialmente a definição de um ADC qual será utilizada ao longo da pesquisa, e posteriormente focando nas ferramentas que estão ou não presentes nesses ambientes. Uma metodologia comparativa foi utilizada, contendo tanto o ponto de vista técnico como o de negócio, de modo que fosse possível identificar as ferramentas disponibilizadas por cada ADC ou repositório utilizado pelas comunidades e portanto fazer a análise do que já existe com as demandas existentes.

Por fim, foi descrita a importância para um ambiente ter sua gestão baseada em processos, qual possibilita a qualquer membro da comunidade envolvido no processo, uma visão global e integrada do fluxo de trabalho, bem como seu total entendimento, independente do local onde são executadas as atividades.

Este capítulo mostrou o embasamento teórico necessário à execução do trabalho de pesquisa. Os resultados obtidos mostram onde estamos contextualizando o presente trabalho e que conceitos foram tomados por base para dar seguimento à pesquisa.

CAPÍTULO 3

O PROJETO VIA DIGITAL

Este capítulo apresenta o projeto Via Digital. Dentro da visão abordada pelo projeto, foi criado um modelo de gestão inovador, potencialmente reutilizável, no todo ou em partes, para qualquer domínio de aplicação. O modelo prevê a auto-organização dos atores das duas comunidades (técnica e de negócio).

3.1 INTRODUÇÃO

Segundo dados do IBGE (2005), cerca de 90% das prefeituras brasileiras dependem quase que exclusivamente de recursos repassados pelo governo federal e estadual. E ainda mostrou que 95,9% destas prefeituras possuem até 100 mil habitantes.

O estudo Softex (2004) sobre aplicação de Software Livre em prefeituras no país verificou que a administração de um bom número de municípios encontra-se debilitada pela ausência de mecanismos de informatização capazes de levar maior eficiência e transparência aos seus processos internos. Foi possível observar, pelo estudo, que prefeituras de todos os tamanhos que adotaram software livre obtiveram ganhos econômicos, melhor aproveitamento de equipamentos, maior autonomia tecnológica, maior transparência de gestão, segurança, independência de fornecedores e estabilidade. Todos os municípios pesquisados julgaram fundamental a existência de um repositório comum, de livre acesso às prefeituras onde estivessem disponíveis informações importantes sobre diversos aspectos da informatização, software livre, avaliações de ferramentas por parte de especialistas, melhores práticas, listas de fornecedores e de profissionais capacitados, etc.

Em algumas áreas, por exemplo, em sistemas operacionais, ambientes de desenvolvimento de software (*JBoss, Eclipse, MySql*), o software livre é presença marcante e novas opções surgem a todo instante. Diferentemente, em outros setores de sistemas de informação, a presença de opções livres é mais rara – e até quase nula (LUCCA,2007).

Esse é o caso dos softwares necessários para a informatização das prefeituras municipais. As regras de negócio e os requisitos não são de domínio dos desenvolvedores de software livre nem há neste setor um contingente de usuários-desenvolvedores capaz de vencer a inércia inicial.

Sobre essa base conceitual e na expectativa de modificar o quadro apresentado, foi proposto o projeto Via Digital (ALVES et. all,2006) financiado pela Finep em colaboração com as seguintes instituições: Universidade Federal de Santa Catarina (UFSC), Centro de Pesquisa Renato Archer (CenPRA), Universidade Federal de Campina Grande (UFCG) e a Sociedade Softex. O Via Digital visa à criação de uma alternativa para a informatização de pequenos municípios, apoiada em pilares não

somente técnicos (como a componentização), mas também estratégicos e políticos, como a sensibilização de prefeituras, desenvolvedores e academia.

A proposta do Via Digital é o fomento ao surgimento de um ecossistema em torno de componentes de software livre destinados à gestão de municipalidades. Este ecossistema envolve prefeituras, desenvolvedores e empresas de software e serviços, universidades e outras instituições de apoio; que se interrelacionam através de modelos de interação e de negócios próprios do domínio e da dinâmica do software livre (MOURA,2006).

Componentes de software apresentam inúmeras definições na literatura. De modo geral, o conceito é definido como uma abstração para a construção de sistemas que pode ser adaptado sobre diversos paradigmas diferentes. Brown e Wallnau (BROWN,1996) descrevem um componente como *"uma não-trivial, quase independente, e substituível parte de um sistema que cumpre uma função clara no contexto de uma arquitetura bem definida"*. Em muitos sentidos, esta descrição é similar a de um objeto em programação orientada a objeto. Componentes possuem uma interface. Eles empregam regras de herança. Mas a definição é levada ainda além. Componentes são definidos para oferecer serviços. Brown e Wallnau descrevem um componente de software como *"uma unidade de composição contratualmente especificada e somente com dependências contextuais explícitas"*.

Com base nas definições estudadas, o projeto Via Digital caracteriza um componente de software como: Um pacote coerente de software que pode ser independentemente desenvolvido e distribuído como uma unidade de Software Livre, que pode ser composto com outros componentes para construir algo maior. São classificados de acordo com seu uso e especificidade:

- ❖ Componente Genérico: são aqueles de uso comum em muitos sistemas, tais como os componentes de interface com os usuários (botões, janelas etc)
- ❖ Componentes de serviços: são componentes que fornecem serviços especializados, mas que não são específicos do ponto de vista de domínio de aplicação, como componentes para tratamento de erros em comunicação de dados, criptografia, segurança, geração de gráfico etc
- ❖ Componente de domínio: são componentes específicos para domínios definidos, que implementam regras (desde simples até mais complexas) de negócios, como por exemplo, regras do setor financeiro ou de construção civil.

Além desta caracterização este trabalho também adotará as definições de demandas e serviços que seguem abaixo:

- ❖ Demanda: é o conjunto de necessidades apresentadas pelo usuário de soluções de TI. Em geral, refletem os processos administrativos realizados sem suporte TI, que acarreta em má qualidade dos serviços e pouca racionalização de recursos. Para cada um desses processos nasce uma demanda.
- ❖ Serviço: é uma atividade, função realizada por um *stakeholder* (por exemplo, comunidade de negócio, comunidade técnica).

Além disso, o ecossistema proposto pelo Via Digital deve adotar um modelo de gestão capaz de promover a interação da comunidade de Software Livre envolvida, assim como sua sustentabilidade. Visando o cumprimento desta meta, foi realizado um estudo bibliográfico sobre os modelos de negócios adotados no mundo aberto, visando obter uma melhor sustentação para o modelo gerado pelo Via Digital.

3.2 MODELO DE NEGÓCIO

Em Krishna (2005) afirma-se que a maioria das companhias deseja maximizar seus lucros, e portanto estão sempre buscando novas formas de gerar receita e reduzir os custos. É crescente o número de companhias que utilizam software livre como uma estratégia para alcançar esses objetivos. Sobre a visão da redução dos custos, desenvolvedores de software podem incorporar o código fonte de um produto com código aberto em um código base existente. Isto permite reduzir o custo de produção pelo reuso do código existente.

Em relação ao aumento de receita, Krishna (2005) afirma que alguns dos produtos SL geram uma demanda bem maior sobre serviços de suporte para as empresas clientes. Esses serviços de suporte incluem instalação, treinamento/ certificação e assistência técnica contínua.

Do ponto de vista do cliente, softwares livres são atrativos devido sua redução nos custos e com desempenho comparável a outros produtos não livres. Instituições governamentais, por exemplo, são altamente motivadas à adoção de SL devido ao nível reduzido de investimentos em TI. Igualmente para usuários individualmente, SL estão se tornando mais acessíveis, diversas aplicações já estão disponíveis para *desktop*. O

Source Forge é um exemplo dessa vertente. Em maio de 2006, este possuía aproximadamente 128 mil projetos cadastrados e 1,3 milhão usuários, hoje esses números ultrapassam 150 mil projetos e mais de 1,5 milhão de usuários.

Em (Krishna (2005) são apresentados modelos de negócio sobre a filosofia software livre, tais como os modelos “O distribuidor”, “Desenvolvedor de Software (Modelo GPL)”, “Desenvolvedor de Software (Modelo não-GPL)” e “Fornecedor de Serviço”. Algumas companhias se beneficiam da venda de hardware que executam software livre, e similarmente, o mercado pelos diversos produtos que podem ser derivados.

O distribuidor

O distribuidor possui acesso tanto ao código fonte como ao software. O distribuidor pode gerar receita das seguintes formas:

- ❖ Provendo o produto em CD – a maioria das pessoas não se sente confortáveis em ter de fazer o *download* do produto a partir de um site na *web*.
- ❖ Provendo serviços de suporte para as empresas clientes – empresas estão dispostas a pagar ao responsável final. Quando algum problema surge, estas não enviam *email* para a lista da comunidade responsável pelo produto, portanto há receita nos serviços, tais como suporte, instalação e treinamento.
- ❖ Serviços de atualizações – no qual empresas podem fazer acordos a longo prazo com distribuidores para obter a última versão e realizar as atualizações.

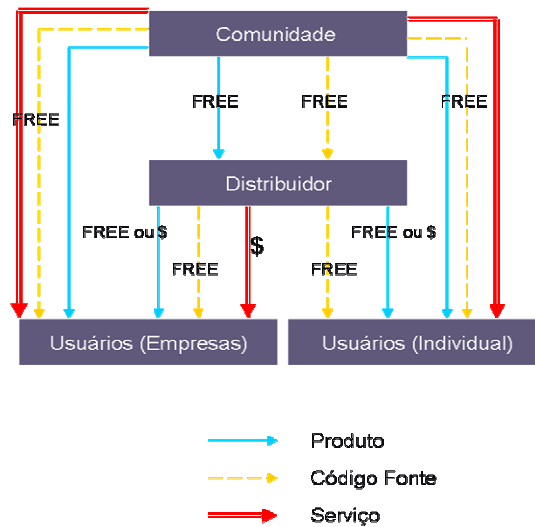


Figura 6 - Modelo de Negócio Distribuidor (KRISHNA,2005).

Desenvolvedor de Software (Modelo Não GPL)

Desenvolvedores de software podem beneficiar-se a partir da comunidade de software livre de duas formas. A primeira, eles podem incorporar o código fonte de um produto existente em um código maior ou ainda criar um novo produto. Segundo, eles podem utilizar todo o produto SL e empacotar com produtos existentes, criando assim um produto derivado.

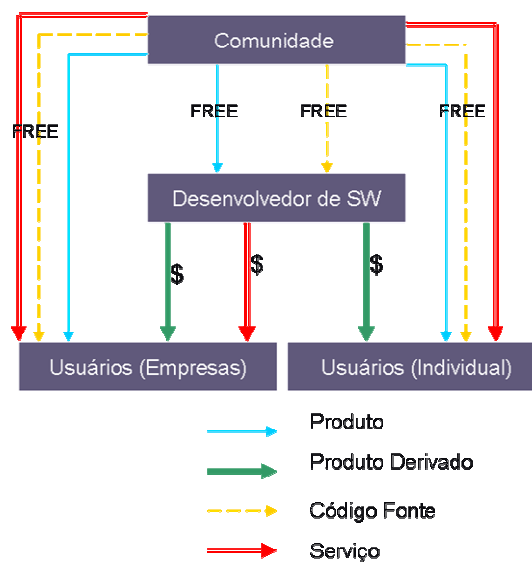


Figura 7 - Modelo de Negócio Desenvolvedor de SW (Modelo Não-GPL) (KRISHNA,2005).

Desenvolvedor de Software (Modelo GPL)

A diferença deste modelo para o anterior é que o desenvolvedor de software é forçado a disponibilizar o código fonte do produto derivado. O modelo GPL impulsiona as inovações, devido ao rápido retorno que é dado pela comunidade de usuários. No entanto a maior diferença se encontra sobre a visão do usuário. O usuário GPL se capacita de forma a poder se engajar em uma comunidade. Já, os usuários não-GPL estão apenas interessados em usar o software.

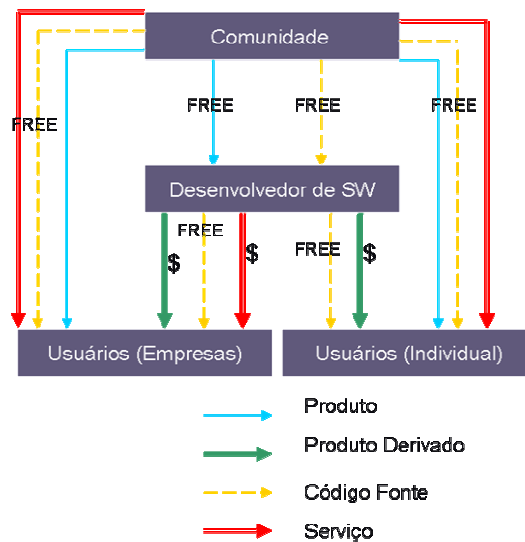


Figura 8 - Modelo de Negócio Desenvolvedor de SW (Modelo GPL) (KRISHNA,2005).

Fornecedor de Serviços

Os fornecedores de serviços não estão interessados como o produto foi adquirido, ou se o produto está sendo usado segundo algum critério, vendem serviços apenas. Eles possuem uma única fonte de receita – serviços.

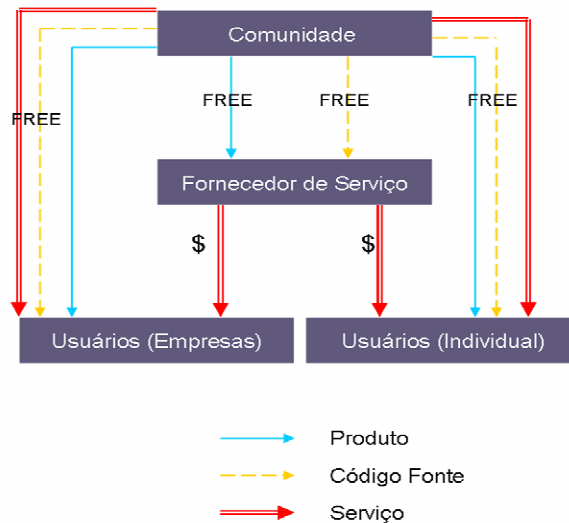


Figura 9 - Modelo de Negócio Fornecedor de Serviços (KRISHNA,2005).

Em Krishna (2005), vender software apenas é insuficiente para sustentar um negócio. O correto é ter software e serviço. Muitos comerciantes de software, já possuem um relacionamento com empresas clientes, assim os comerciantes usam o serviço como forma potencial de rendimento secundário.

3.3 MODELO DE NEGÓCIO VIA DIGITAL

O modelo proposto Moura (2006) é baseado em um conceito de *framework* de processos de gestão do conhecimento sobre Software Livre para o domínio de aplicação (processos-fim), e em uma rede de cooperação cíclica produtiva. Esta rede se diferencia dos *clusters* convencionais por incluir o ator “desenvolvedor de software livre” e uma instituição que orchestra (processo-meio) a integração entre dois mundos diferentes: comunidade de Software Livre e comunidade de empreendedorismo.

O modelo é composto por processos núcleos, que compreendem três áreas: marketing, financeiro e técnico. Além destes, existem os processos de orquestração, qual são responsáveis pelo pleno funcionamento das áreas citadas. A organização dos macros pode ser observada na Figura 10. No apêndice são detalhados todos os macros processos em sub-processos.



Figura 10 - Modelo de negócio Via Digital.

O ambiente proposto nesse trabalho foi implementado baseado no ciclo de vida do macro processo “Desenvolvimento de SL e Garantia da Qualidade”, como forma de permitir a geração de uma comunidade de software livre, motivada a contribuir com o desenvolvimento de componentes livres para atender uma demanda do domínio da aplicação. A auto-sustentação do ambiente é promovida pelo sub-processo Gerência de Concurso, pertencente ao macro processo Empreendedorismo.

3.4 CONSIDERAÇÕES FINAIS

Neste capítulo foi descrito o projeto Via Digital, em qual este trabalho esteve inserido. Durante a realização deste projeto, o núcleo da Universidade Federal de Campina Grande (UFCG), propôs um modelo de gestão que visa a criação e consolidação do empreendedorismo local através do desenvolvimento de soluções livres. Pretendendo-se dar sustentação ao modelo proposto foi realizado um estudo sobre os modelos de negócio existentes envolvendo o mundo aberto.

Em seguida, é proposta uma implementação do ambiente definido pelo modelo Via Digital. O ambiente possui ênfase nos processos Desenvolvimento de SL e Garantia da Qualidade e Empreendedorismo.

CAPÍTULO 4

FLOAPP

Este capítulo discorre sobre ambiente de desenvolvimento proposto, o FLOApp - Free/Open/Livre Aplicações apresentando como as comunidades (de negócio e técnica) interagem, descrevendo os serviços oferecidos por cada uma desta. É explanada ainda a arquitetura proposta, que visa simplificar a adição de novas ferramentas ao ambiente e a satisfação dos usuários. Também é apresentada a caracterização do FLOApp segundo a modelagem dos processos de negócio responsáveis por toda orquestração do ambiente.

4.1 INTRODUÇÃO

A proposta do FLOApp (Free/Open/Livre Aplicações) é a criação de um serviço completo, operacional e auto-sustentável que integre uma biblioteca de componentes livres e que sirva de elo entre a comunidade de negócio (CN), organizada em torno do modelo de negócio Via Digital (MOURA,2006), e a comunidade técnica (CT).

O FLOApp será ponto de referência para abastecer a CN com software pré-formatado para as necessidades mais comuns e componentes genéricos para montagem de sistemas mais específicos, de acordo com características e requisitos específicos dos clientes. O serviço provido também será uma referência para busca e difusão de informações sobre a dinâmica do Software Livre.

Para viabilizar tais objetivos, o FLOApp disponibiliza meios para que a CN localize e obtenha tanto componentes prontos para o atendimento de necessidades, como componentes que sejam utilizados no desenvolvimento de novas soluções. Ainda são disponibilizadas ferramentas que facilitam o gerenciamento dos projetos de componentes mantidos, de forma a incentivar a participação da CN no desenvolvimento e na melhoria do repositório.

O FLOApp também oferece um meio efetivo de visibilidade a novos projetos de componentes voltados ao cliente, expondo-os diretamente ao seu público-alvo e facilitando seu sucesso e fortalecimento.

Um projeto no FLOApp significa a centralização do esforço de desenvolvimento em torno de um componente, tendo um administrador central, provavelmente o criador do projeto, e outros sub-administradores designados por este, com algumas permissões especiais, a exemplo de alteração de conteúdo do sítio do projeto e admissão de novos desenvolvedores. O repositório de componentes é formado pelo conjunto dos componentes oferecidos pelos projetos mantidos, em que cada componente tem dinâmica independente dos outros desenvolvidos.

A interação entre as comunidades envolvidas e os serviços disponíveis para cada uma são mostrados na Figura 28.



Figura 11 – Visão Geral do FLOApp

4.1.1 COMUNIDADE DE NEGÓCIO (CN)

Dentro do ambiente que está sendo proposto, é fundamental a participação da comunidade de negócio, até então, não existente no mundo aberto. Isto pode ser observado, principalmente, devido ao desenvolvimento não ser orientado a produção de sistemas de informação. Em que existem regras de negócio envolvidas e se faz necessário uma maior compreensão dos requisitos envolvidos por parte dos desenvolvedores. A interação entre o cliente (CN) e a comunidade técnica, visa promover uma rede de colaboração, seja para qualificar os desenvolvedores nas tecnologias, processos, na melhor forma de integrar o cliente ao ciclo de desenvolvimento, como também a própria comunidade de negócio ter um referencial para demonstrar suas necessidades e interagir entre si, criando novas oportunidades de negócio.

A comunidade de negócio, por fim, representa os interesses dos usuários envolvidos com o domínio da aplicação. Nesta comunidade os usuários tanto podem consumir os serviços oferecidos pela comunidade técnica como disponibilizar serviços a serem

consumidos, seja pela própria comunidade de negócio como pela CT. A seguir são descritos os serviços consumidos pela CN conforme ilustrado na figura 28.

Rastrear componentes

A CN pode rastrear os componentes que estão disponibilizados no repositório com estado atual “Produção”. Por conseguinte, somente após a CT liberar o *release* e alterar o estado do componente este poderá ser rastreado utilizando palavras chaves e categoria (Figura 29). Através deste serviço a comunidade de negócio poderá localizar os componentes que tenha interesse, realizar o *download* e obter informações sobre o uso, atividade e qualidade do componente. Caso nenhum seja encontrado, poderá requisitar uma demanda ou contribuir com uma já existente.



Figura 12 - Rastrear componentes.

Visão geral do projeto

Este serviço visa fornecer as informações necessárias para auxiliar a tomada de decisão da comunidade de negócio a cerca do projeto selecionado (Figura 30). As informações disponíveis são:

- ❖ Descrição do projeto;
- ❖ Responsáveis pelo projeto;
- ❖ O último *release* liberado disponível para *download*;
- ❖ O grau de atividade;
- ❖ Os usuários mais ativos;
- ❖ A frequência de acesso;
- ❖ A qualificação;
- ❖ A lista dos usuários que realizaram os últimos *downloads*;
- ❖ A lista de demandas correspondente ao projeto.

The screenshot displays the FLO App interface. At the top, there's a navigation bar with 'Home' and 'Logout' links. Below it, a central banner for 'FLO App' features a circular arrow graphic and labels for 'Componente' and 'Demanda'. The main content area is divided into three columns. The left column lists four project items, each with a date '07/08/09' and a title: 'ReCriAd: registro de crianças e jovens - A Rede. 06/2007', 'Mais cidades na Infovia carioca - A Rede. 06/2007', 'Intercambio Latino Americano de FLOSS - A Rede. 06/2007', and 'Pedreira (SP) Inaugura Infovia Digital'. The middle column is titled 'ViaCadim' and contains a description: 'O ViaCadim é um componente que faz parte do projeto Via Digital direcionado à área de cadastro de imóveis.' It also lists 'Categoria: Administração' and 'Idioma: Português'. Below this, it shows 'Registrado: 07-08-2007 18:40', 'Percentual de Atividade: 50%', and a link to 'Ver estatísticas de atividade do projeto.' A 'Download' button is positioned at the bottom of this section. The right column, titled 'Informações dos Desenvolvedores', lists 'Administradores de Projeto: Ana Isabella Muniz' with a '[Ver Membros]' link and a 'Participar' button. Below that, the 'NOTA do Projeto' section shows 'Média: 8,76' and 'Votos: 9'. It includes a dropdown menu for 'Qual a nota desse projeto:' with '1-Péssimo' selected and an 'enviar' button.

Figura 13 - Visão geral do projeto.

Download do componente

A CN pode realizar o *download* dos *releases* dos componentes disponíveis no repositório com status “Produção” (Figura 31). Após o *download*, o usuário é convidado a qualificar o componente segundo sua utilização, documentação e sobre as informações dispostas no FLOApp.



ViaCadim

O ViaCadim é um componente que faz parte do projeto Via Digital direcionado à área de cadastro de imóveis.

Uploads	Testes
Uploads: 12	Uploads: 3
Último: 2007-08-07 - 18:23	Último: 2007-08-07 - 20:12
Penúltimo: 2007-08-07 - 10:14	Penúltimo:

Últimas versões de arquivos

Pacote	Versão	Data	Notas/Monitorar	Download
viacadim-beta	war	2007-08-07	 - 	Download
viacadim-v0.1	war	2007-08-07	 - 	Download

[Ver todos os arquivos do projeto]

Figura 14 - Download do componente.

Qualificação do componente

A CN após realizar o *download* do componente, este poderá ser qualificado de acordo com as seguintes visões:

- ❖ Instalação
- ❖ Configuração
- ❖ Utilização
- ❖ Documentação
- ❖ Suporte (através das listas e fóruns de discussões)

- ❖ Conformidade com o problema solucionado.

Cadastrar demanda

A comunidade de negócio pode cadastrar uma nova demanda, alterar uma demanda criada pelo mesmo usuário ou então poderá removê-la (Figura 32). Para requisitar uma nova demanda é necessário informar:

- ❖ Descrição – apresenta os casos de uso, requisitos do componente;
- ❖ Categoria – dentro do domínio da aplicação, deve ser selecionada a categoria que melhor representa a demanda;
- ❖ Prioridade – corresponde ao grau de necessidade e importância que o componente representa para a comunidade de negócio;
- ❖ Responsável – o FLOApp associa o usuário que está criando a demanda como responsável, administrador da demanda. Possui portanto, os privilégios de alterar e remover a demanda;
- ❖ Status – corresponde ao estado atual da demanda. Inicialmente seu status igual a “Análise”, a partir das contribuições vindas da comunidade seu status poderá ser modificado.

The image shows a web application interface for 'FLO App'. At the top, there is a navigation bar with 'Home' and 'Logout' links. Below this, a central banner displays the 'FLO App' logo with a circular arrow graphic and the text 'Componente Demanda'. The main content area is titled 'Cadastrar Demanda' and contains a form with the following fields: 'Nome:' (text input), 'Prioridade:' (dropdown menu set to '3-Média'), 'Status:' (dropdown menu set to 'Análise'), 'Categoria:' (dropdown menu set to 'Administração'), 'Descrição:' (text area), and 'Arquivo:' (file upload field with a 'Browse...' button). A red error message is visible below the 'Categoria:' field: '* Coloque a descrição da demanda ou faça o upload de um arquivo'. On the left side, there is a sidebar with a list of existing demands, each with a date '07/08/09' and a title: 'ReCriAd: registro de crianças e jovens - A Rede, 06/2007', 'Mais cidades na Infovia carioca - A Rede, 06/2007', 'Intercambio Latino Americano de FLOSS - A Rede, 06/2007', and 'Pedreira (SP) Inaugura Infovia Digital'.

Figura 15 - Cadastrar demanda.

Rastrear demanda

A comunidade de negócio pode rastrear demandas cadastradas no repositório, através de palavras chaves ou categorias.

Cooperar com uma demanda

A CN pode cooperar com uma demanda existente no repositório, adequando ela à novas necessidades do usuário. Este poderá inserir novos requisitos, casos de uso, como também mediante a importância desta alterar sua prioridade e ainda seu estado, fazendo com que essa possa ser visualizada pela CT.

Alterar o status da demanda

A comunidade de negócio pode alterar o estado de uma demanda desde que esta esteja completa o suficiente e já atenda às necessidades do usuário em questão. Uma demanda poderá ter seu estado igual à “Análise” quando apenas pode ser visualizada pela CN e está na fase de construção da demanda (especificação do componente), isso devido a comunidade de negócio não ter definido e formatado qual realmente é sua necessidade. A partir do momento que algum integrante da CN acha que uma demanda corresponde a uma precisão instantânea, então o status é alterado para “Produção”. Neste a demanda já pode ser observada pela comunidade técnica.

4.1.2 COMUNIDADE TÉCNICA (CT)

A comunidade técnica representa as empresas de Tecnologia de Informação, comunidades de software livre, estudantes e desenvolvedores interessados em contribuir com algum projeto. Assim como a comunidade de negócio, a CT também oferece serviços que são consumidos pelo cliente (CN) descritos na seção anterior. A seguir são descritos os serviços consumidos pela CT, conforme ilustrado na figura 28.

Rastrear demanda

A comunidade técnica pode rastrear demandas cadastradas no repositório com estado igual a “Produção”, através de palavras chaves ou categorias.

Visão geral de uma demanda

Este serviço visa fornecer as informações necessárias para auxiliar a tomada de decisão da comunidade técnica a cerca de uma demanda selecionada (Figura 23). As informações disponíveis são:

- ❖ Descrição – contendo todos os casos de uso do componente, descritos pela CN;
- ❖ Categoria;
- ❖ Prioridade;
- ❖ Responsável – através deste a CT poderá interagir com a CN, para esclarecer dúvidas sobre os casos de uso e dar sugestões;

Componente Demanda

Informações dos Usuários	Nome: Controle da Merenda Escolar
Administradores de Demanda: PM Fagundes	Categoria: Educação
[Ver Membros]	Prioridade: 4
Estatísticas	Responsável: PM Fagundes
NOTA da Demanda	Descrição:
Média: 0,00	US: controlar a aplicação dos recursos destinados à merenda escolar;
Votos: 0	US: promover a elaboração dos cardápios dos programas de alimentação escolar em parceria com nutricionista.
Qual a nota dessa Demanda: 1-Péssimo	US: avaliar os mapas de controle da merenda escolar das unidades escolares.
enviar	

20/07/07
[CoreBuilder, uma revolução no desenvolvimento de aplicativos](#)

20/07/07
[48% das empresas brasileiras programam em java](#)

25/04/07
[Red Hat compra empresa para integrar a JBOSS](#)

16/04/07
[Sun firma parceria para distribuir Solaris em note book](#)

Figura 16 - Visão geral de uma demanda.

Qualificar demanda

Após a comunidade técnica realizar o download de uma demanda, poderá qualificá-la de acordo com as seguintes visões:

- ❖ Documentação;
- ❖ Conformidade com o problema;
- ❖ Viabilidade;
- ❖ Suporte.

Este serviço visa auxiliar a CT a tomar decisões sobre uma determinada demanda. Ao apresentar, por exemplo, uma baixa qualificação em suporte da comunidade de negócio, a comunidade técnica saberá desde o início da dificuldade que terá na mitigação dos riscos e problemas acerca do desenvolvimento do componente.

Responsabilizar pela demanda

Qualquer integrante da comunidade técnica pode ser responsável pelo desenvolvimento de uma demanda. Para isso deve criar um novo projeto e associar a demanda escolhida ou associar a um projeto já existente que apresente os requisitos expressados na descrição da demanda. Ao se tornar responsável desta, a comunidade técnica poderá interagir com a CN, se necessário, para obter mais informações ou esclarecimentos de dúvidas que venham a surgir durante todo o ciclo de desenvolvimento.

Criar novo projeto

Uma vez um integrante da comunidade técnica esteja cadastrado no FLOApp, poderá criar novos projetos associados a demandas ou a partir de uma oportunidade de negócio identificada por ele. Logo, este integrante será considerado o administrador desse projeto, responsabilizando-se por todas as decisões de desenvolvimento e ainda dará suporte a CN que venha se interessar pelo componente resultante do projeto.

Cooperar com um projeto existente

Qualquer integrante da CT interessado em contribuir com algum projeto já cadastrado no FLOApp poderá se associar a esse e cooperar com os responsáveis pelo projeto seja no desenvolvimento das atividades ou com *feedbacks*.

Desenvolver/Testar/Documentar componente

A comunidade técnica é responsável pelas atividades de desenvolvimento, teste e documentação dos componentes. O FLOApp fornece um repositório para que estas atividades possam ser realizadas de forma cooperativa.

Alterar o estado do componente

Ao ser criado um projeto no FLOApp, o status inicial é configurado para “Análise”, desta forma apenas a comunidade técnica pode visualizar este projeto. Após a CT começar a interagir entre si e as contribuições serem adicionadas o status poderá ser alterado para “Desenvolvimento”. E ao ser lançado um release, seu status é alterado para “Produção” e então comunidade de negócio também terá acesso ao projeto .

4.2 ARQUITETURA DO FLOAPP

A arquitetura do FLOApp foi projetada visando prover interoperabilidade, flexibilidade e reuso. Na Figura 34 segue o projeto arquitetural desenvolvido, ilustrando a interação entre os usuários e os módulos do sistema, bem como, os protocolos de comunicação. O projeto foi baseado nos requisitos não funcionais levantados, descritos a seguir.

RNF01: Usabilidade

A interface com o usuário é de vital importância para o sucesso do FLOApp. Principalmente por promover um ambiente de interação entre usuários sem conhecimento prévio em TI e estes não possuem tempo disponível para aprender como utilizar o ambiente.

RNF02: Controle de Acesso - Segurança

A função só pode ser acessada por usuário com perfil de negócio ou técnico.

RNF03: Desempenho

Embora não seja um requisito essencial ao sistema, deve ser considerada por corresponder a um fator de qualidade de ambiente.

RNF04: Arquitetura baseada em *plugins*

Segundo Birsan (2005), Arquiteturas baseadas em *plugins* são caracterizadas por possuir um núcleo funcional, no qual extensões podem ser adicionadas dinamicamente. Logo, o núcleo funcional do Floapp possui um conjunto mínimo de funcionalidades necessárias para a interação das comunidades de usuários e a partir da carência seja técnica ou de negócio, *plugins* devem ser incrementados ao ambiente, adicionando-lhe as novas funcionalidades.

RNF05: Uso de Padrões

É um requisito importante para a manutenção dos módulos que compõem o ambiente,

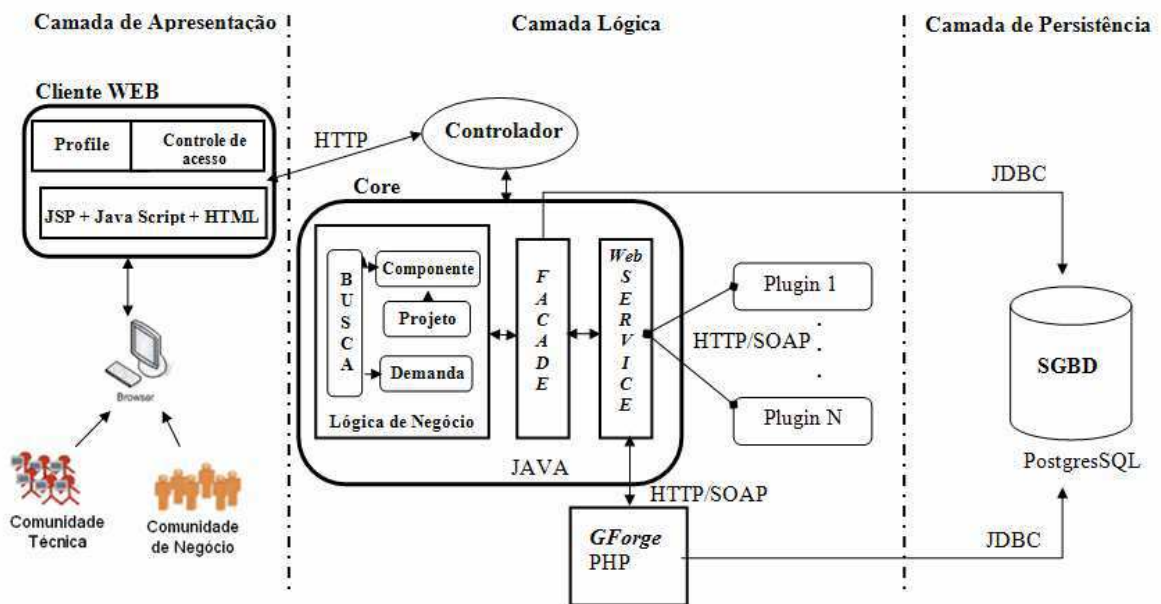


Figura 17 - Projeto Arquitetural do FLOApp.

4.2.1 CAMADA DE APRESENTAÇÃO

Nesta camada um *browser web* é utilizado para implementar a interação entre as comunidades e o FLOApp. Toda informação de entrada é validada e enviada para a camada lógica através do cliente *Web*, qual é baseada em *Java Server Pages* (JSP) e HTML.

4.2.2 CAMADA LÓGICA

Esta camada representa a lógica do FLOApp. Nesta fazemos uso do padrão de projeto Facade (GAMMA, 2000) para separar a lógica de negócio da camada de persistência. Essa Facade é utilizada também pelo módulo *Web Service* qual gerencia toda a comunicação realizada entre o núcleo da aplicação e o repositório, neste caso foi utilizado o GForge (GFORGE,2007), devido ser um projeto de software livre baseado no sistema original do Source Forge. Todavia qualquer outro que implemente os serviços descritos pelo módulo poderá ser utilizado. Da mesma forma, novas ferramentas (*plugins*) podem ser adicionadas ao FLOApp.

Um *plugin* pode ser criado de acordo com as necessidades de cada comunidade. O objetivo é tornar simples a integração de novas ferramentas ao FLOApp à medida que estas necessidades aumentem, seja do lado da comunidade de negócio ou técnica. Para adicionar um *plugin*, este deverá estar no mesmo servidor de aplicações do FLOApp, em intervalos de cinco minutos o servidor verifica a existência de um novo projeto e então o adiciona ao FLOApp.

Um exemplo de um *plugin* foi criado, o FLOAppTest (COITINHO et. all, 2007), uma ferramenta para desenvolvimento e execução de testes de aceitação automático. Esta foi desenvolvida devido à carência de uma ferramenta de teste mais próxima dos clientes nos ambientes colaborativos existentes.

Na lógica de negócio são apresentados quatro principais módulos:

- ❖ Busca: este módulo é responsável pelo rastreamento de componentes realizado pela comunidade de negócio, apresentando como resultado uma visão geral do projeto correspondente ao componente pesquisado. É também realizado o rastreamento de demandas, tanto pela comunidade de negócio como pela comunidade técnica, para esta são apresentadas apenas as demandas com estado de finalizada.

- ❖ **Componente:** este módulo apresenta as atividades relativas ao download e qualificação de componentes. Cada componente está associado a um projeto dentro do repositório.
- ❖ **Projeto:** este módulo é responsável por criação de projetos ou recuperação de já armazenados no repositório e apresentá-los para comunidade.
- ❖ **Demanda:** este módulo corresponde às atividades de cadastro, cooperação, qualificação, priorização de demandas.

4.2.3 CAMADA DE PERSISTÊNCIA

Esta camada representa a forma como os dados do FLOApp são armazenados. O banco de dados utilizado é o PostgreSQL. Todavia, esta camada é flexível para acomodar qualquer sistema de banco de dados que possa ser acessado via *driver* JDBC.

4.2.4 LIMITAÇÕES DA ARQUITETURA

Na arquitetura projetada, é possível utilizar qualquer repositório, entretanto uma vez adicionado ao FLOApp, este dependerá das informações dos projetos presentes no repositório. A comunicação entre esse e o FLOApp ocorre através de um *web service*, qual coleta, trata e envia as informações do repositório para o ambiente. Portanto, caso haja o interesse em trocar de repositório, poderá haver perdas de informações sobre componentes e projetos.

4.3 CARACTERIZAÇÃO DO FLOAPP

O modelo de gestão do Via Digital propõe através do processo “Desenvolvimento de SL e Garantia da Qualidade” prevê a criação de um ambiente auto-sustentável que auxilie a promoção do empreendedorismo e também ofereça suporte ao desenvolvimento de componentes livres e serviço para um domínio de aplicação específico, assegurando e certificando a qualidade de ambos.

Além disto, segundo proposto por Moura (2006), o ambiente deve fornecer suporte, ferramentas e conjunto de práticas, plataformas para desenvolvimento e execução, arquiteturas de aplicações semelhantes, interface com usuário, interoperabilidade e requisitos de segurança, documentação, padrões de codificação e *seed code* de kits de aplicações e infra-estrutura certificadas. Suporte (treinamento, consultoria, *expertise* no desenvolvimento tecnológico) e acompanhamento dos serviços disponibilizados.

De acordo com o fluxograma do desse macro processo (ver Apêndice), o fluxo das atividades disponíveis pelo FLOApp pode ser observado na Figura 2. O ambiente é provocado a partir do surgimento de demandas pela comunidade de negócio. Esta em colaboração, entre os integrantes, devem gerar a especificação contendo os requisitos desejados a serem implementados pela comunidade técnica. E ao final do processo de desenvolvimento, o componente de gestão será disponibilizado.

O ambiente não fornece apenas suporte a requisição de demandas e desenvolvimento do componente de forma isolada, mas, principalmente, torna possível a interação entre as comunidades durante todo o fluxo do processo. Visto que a natureza das soluções desenvolvidas envolve regras de negócio, quase sempre desconhecidas pelos desenvolvedores.

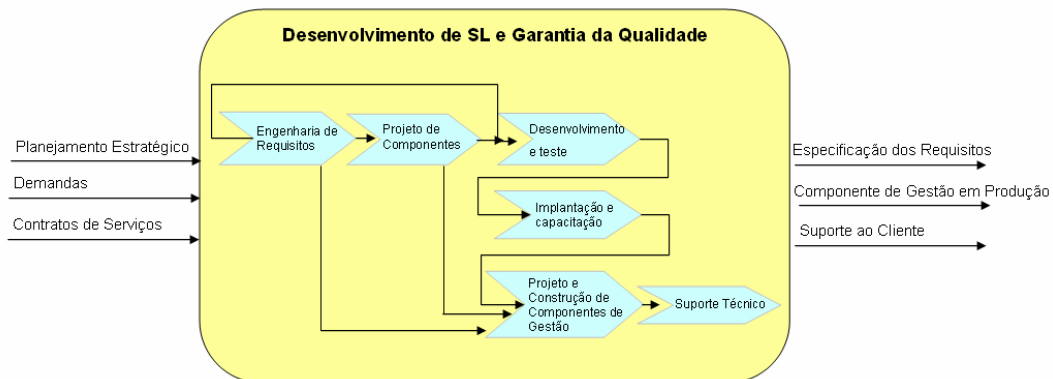


Figura 18 - Fluxo geral do FLOApp baseado no processo "Desenvolvimento de SL e Garantia da Qualidade".

A especificação do FLOApp é corresponde a um conjunto restrito das principais atividades do ciclo de desenvolvimento de um componente, Estas atividades compreendem desde o processo “Engenharia de Requisitos”, qual possui a gestão de demandas como atividade central, até o processo “Suporte Técnico”, realizado pelo

suporte fornecido aos usuários de negócio pela comunidade técnica através de listas de discussões, fóruns ou *emails*.

Entretanto, nem todos os processos foram totalmente contemplados, devido o FLOApp fornecer um conjunto funcional mínimo e a partir das necessidades das comunidades, extensões podem ser adicionadas a ele. O Quadro 21 apresenta as contribuições do ambiente que foram implementadas no FLOApp.

Requisitos	FLOApp
RF1 – Ambiente de desenvolvimento integrado	x
RF2 – Gestão de demandas	x
RF3 – Prototipação	
RF4 – Gestão de configuração	x
RF5 – Rastreamento de falhas e/ou melhorias	x
RF6 – Testes	x
RF7 – Comunicação	x
RF8 – Documentação	x
RF9 – Gestão de tarefas	x

Quadro 3 - Contribuições do FLOApp.

4.4 CONSIDERAÇÕES FINAIS

Neste capítulo foi descrito o ambiente de desenvolvimento de componentes de gestão, qual visa promover e facilitar a interação entre as comunidades técnica e de negócio. O FLOApp foi desenvolvido seguindo a especificação do ambiente proposto pelos processos Desenvolvimento de SL e Garantia da Qualidade e Empreendedorismo, entretanto apenas um conjunto mínimo funcional foi implementado. Para viabilizar o acréscimos das demais atividades sugeridas foi proposta uma arquitetura baseada em *plugins*.

No próximo capítulo, será abordado através de um estudo de caso as contribuições e restrições do uso do FLOApp tendo como comunidade técnica estudantes e professores universitários e como comunidade de negócio as prefeituras.

CAPÍTULO 5

ESTUDO DE CASO

Este capítulo apresenta a análise dos resultados obtidos com a utilização do FLOApp em um estudo de caso para validação do ambiente.

5.1 CONSIDERAÇÕES INICIAIS

No Capítulo 3, foi apresentado um modelo de gestão visando promover a interação da comunidade de Software Livre com a comunidade de Empreendedorismo. A partir do fluxo dos processos que compõem esse, foi especificado um ambiente para fornecer suporte ao desenvolvimento de aplicações do domínio. Logo, visando validar a especificação gerada e assegurar os objetivos deste trabalho (definidos no Capítulo 1), o estudo de caso ora descrito tem por objetivo investigar a eficiência do FLOApp enquanto ambiente de suporte a inserção da comunidade de negócio no desenvolvimento de componentes livres.

Estaremos analisando o cumprimento dos seguintes objetivos:

- ❖ Permitir o rastreamento de falhas
- ❖ Permitir a integração de código
- ❖ Permitir a criação e execução de testes
- ❖ Fornecer métricas de documentação
- ❖ Fornecer métricas de reuso
- ❖ Permitir as comunidades técnicas e de negócio a qualificação de demandas e componentes respectivamente
- ❖ Permitir a gestão de mudanças
- ❖ Privacidade e Segurança
- ❖ Permitir o rastreamento de componentes
- ❖ Permitir a gestão de demandas
- ❖ Fornecer indicadores de satisfação
- ❖ Fornecer uma FAQ (Dúvidas mais frequentes)

Portanto, para tal investigação este estudo de caso foi implementado em duas etapas. Em cada etapa foi realizado um concurso de Software Livre. Primeiramente foi realizado um concurso piloto, envolvendo uma instituição de ensino e outro em âmbito regional, ambos promovidos pelo projeto Via Digital, com o objetivo geral de fomentar a qualificação dos estudantes em engenharia de software e a produção de componentes livres, que atendam as demandas levantadas pela comunidade de negócio, associado as práticas de desenvolvimento de Software Livre como mostrado no modelo de iteração do concurso (Figura 35).



Figura 19 - Modelo de interação do concurso. (MUNIZ,2007)

5.1.1 IDENTIFICANDO AS COMUNIDADES DO FLOAPP

A comunidade de negócio é formada pelas prefeituras de Fagundes e Patos. Nos Quadros 22 e 23 são apresentadas as informações básicas de cada prefeitura.

População	Densidade Populacional	PIB per capita	IDH	Colégio Eleitoral
10.929	67,4 hab./km ²	R\$ 2.253,11	0,559	9.670

Quadro 4 - Informações básicas da cidade de Fagundes, Paraíba - Brasil (IBGE,2006).

População	Densidade Populacional	PIB per capita	IDH	Colégio Eleitoral
99.494	191,6 hab./km ²	R\$ 3.969,00	0,763	62.371

Quadro 5 - Informações básicas da cidade de Patos, Paraíba - Brasil (IBGE,2006).

A comunidade técnica, por sua vez, é formada pelos estudantes e tutores credenciados pelas instituições de ensino superior (IES) participantes dos concursos.

5.1.2 DINÂMICA DO CONCURSO

A dinâmica de cada certame realizado é apresentada na Figura 36. A partir de uma demanda identificada pela comunidade de negócio, a entidade coordenadora (EC) do concurso divulga o edital do concurso (vide anexo A) contendo a especificação do componente a ser implementado.

Os tutores que possuem vínculo com alguma IES credenciada, podem então inscrever seus alunos agrupados em equipes ou os próprios estudantes interessados em participar podem ser inseridos como voluntários por um outro professor que venha ter maior afinidade.

Visando a qualificação e interoperabilidade do componente a ser produzido, a EC disponibiliza para os tutores um kit de tecnologia, contendo um modelo arquitetural seguido por padrões de tecnologias propostas para o desenvolvimento do componente Via Digital. Também é fornecido ao tutor um kit didático, a partir deste pretende-se qualificar os cursos de engenharia de software, disponibilizando *templates* de aulas, listas de exercícios, página *web* para interação entre professor e aluno e ainda livros. O kit de treinamento é fornecido para os tutores, que estão orientando estudantes que não são seus alunos e precisam de um reforço seja em processo de desenvolvimento ou em alguma tecnologia. Este kit dispõe de uma aplicação exemplo, desenvolvida sobre os padrões definidos no kit tecnológico, tutoriais e um guia de execução de práticas, contendo todo o passo a passo de como qualquer integrante da comunidade técnica pode colaborar com o desenvolvimento de um componente, abordando além das atividades gerenciais de projeto, também indica um conjunto de ferramentas e padrões livres que apóiam todo o ciclo de vida do componente.

Ao longo do processo de desenvolvimento os tutores interagem com as equipes, fornecendo-lhes o material disponibilizado e avaliando as contribuições realizadas pelos estudantes. No modelo iterativo e incremental adotado no concurso, as contribuições são avaliadas em ciclos, de acordo com o conjunto de requisitos definido para cada iteração. Ao final, o componente mais pontuado é premiado definindo o ganhador do concurso.

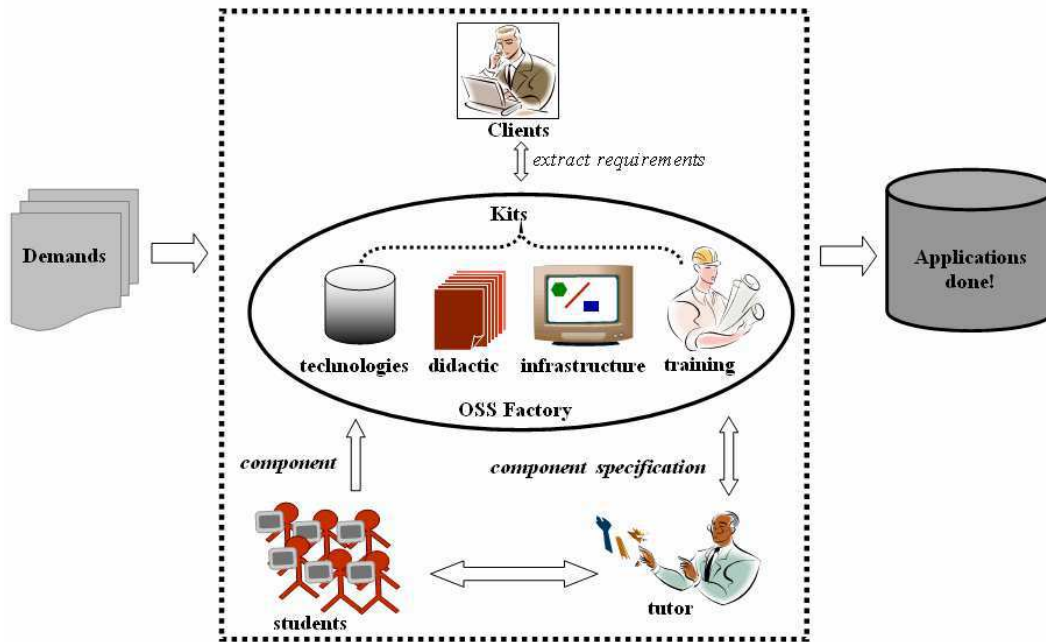


Figura 20 - Dinâmica do concurso (MUNIZ,2007).

5.1.2 IDENTIFICANDO OS VÉRTICES QUE COMPÕEM O CONCURSO

As duas versões do concurso seguem a dinâmica apresentada na seção anterior. Com a experiência e o aprendizado da realização do concurso piloto (CP), o concurso em nível regional (CNR) desejava aumentar o tamanho da comunidade de Software Livre envolvida na produção de um componente de negócio e, diferentemente da primeira fase, fornecer um ambiente que apoiasse a comunidade em todas as etapas do processo de desenvolvimento. Além disso, a execução do CP foi fundamental para auxiliar a especificação dos requisitos do FLOApp, avaliando as atividades que deveriam ter maior foco no seu desenvolvimento. Este ambiente foi utilizado como plataforma de suporte ao concurso, fornecendo aos participantes um ambiente único onde cada equipe pôde cooperar entre si e ao mesmo tempo interagir com a comissão organizadora do concurso, com os tutores e com as demais equipes. Portanto, durante a realização do CNR foi possível avaliar, de forma efetiva, os serviços disponibilizados pelo FLOApp.

O CP foi realizado pela UFCG, tendo como participantes os alunos da disciplina de Laboratório de Engenharia de Software (LES). Esta é uma disciplina orientada à execução de projetos concretos de desenvolvimento de software. Uma das maiores contribuições da disciplina é permitir que as equipes de alunos implantem um processo de desenvolvimento em projetos com clientes reais.

O Quadro 24 apresenta a comunidade técnica envolvida neste certame.

Integrantes	#Total
IES	1
Tutor	1
Estudantes	10
Equipes	5
Componente	1

Quadro 6 - Participantes do Concurso Piloto.

O concurso em nível regional (CNR) foi organizado pela UFCG e envolveu a participação mais outras cinco IES. O Quadro 25 apresenta a comunidade técnica desta etapa do concurso.

Integrantes	#Total
IES	6
Tutor	9
Estudantes	53
Equipes	14
Componente	1

Quadro 7 - Participantes do Concurso Regional.

5.2 EXECUÇÃO DO ESTUDO DE CASO

Como mostrado em Muniz (2007), a dinâmica do concurso envolve a produção de um componente, selecionado de uma lista das principais demandas da comunidade de negócio em um encontro envolvendo seis prefeituras de todo o Brasil (Patos - PB, Recreio - MG, Amparo – SP, Canela - RS, Santa Clara do Sul – RS) realizado durante um evento de Software Livre (SOLISC,2005). Cada prefeitura conceituava cada demanda de acordo com sua prioridade de 1(menor) a 5(maior). Nesta ocasião, as cinco demandas que obtiveram maior pontuação (em ordem decrescente) foram:

1. Controle de Medicamentos e Farmácia
2. Controle Acadêmico
3. Gestão de Imóveis e Contribuintes
4. Gestão de Frota
5. Gestão de Processos e Protocolo

5.2.1 FASE 1 – DEFINIÇÃO DOS REQUISITOS

O concurso piloto teve como alvo o desenvolvimento do componente Controle de Medicamentos e Farmácia, como descrito em Muniz (2007). A partir da demanda levantada os requisitos foram especificados pela entidade coordenadora juntamente com a comunidade de negócio (Prefeitura de Patos). As principais dificuldades enfrentadas nessa fase foram:

- ❖ Com o surgimento de dúvidas relativas ao domínio do problema, o tutor ou os estudantes tinham que se deslocar até a prefeitura e localizar o responsável para sanar os problemas;
- ❖ Os estudantes ainda não têm maturidade na abordagem com o cliente para coletar os requisitos;
- ❖ Em diversas situações, mesmo as pessoas que detêm expertise nem sempre conseguem informar o que realmente desejavam.

O CNR teve como alvo o desenvolvimento do componente Gestão de Imóveis e Contribuintes. A entidade coordenadora levantou em parceria com a comunidade de negócio (Prefeitura de Fagundes) a especificação dos requisitos. Esta foi lançada juntamente com edital do concurso no ambiente de suporte ao concurso. A principal dificuldade enfrentada pelos desenvolvedores, nessa fase também foi a compreensão adequada do domínio do problema. O FLOApp, no entanto, fornece ferramentas para troca de informações, dúvidas e sugestões. À medida que os problemas surgiam, as equipes e tutores interagiam com a EC através dos fóruns e listas de discussões visando saná-los. Todavia, apenas quando um especialista da área do negócio começou a interagir no fórum e então foi elaborado o documento de casos de uso em conjunto com a comunidade, essa dificuldade foi superada.

5.2.2 FASE 2 – DESENVOLVIMENTO E TESTES

No CP, as equipes trabalharam de forma isolada e descentralizada. O tutor era o referencial para todas as equipes, tendo assim a responsabilidade de centralizar as informações em uma página da disciplina. Os materiais de suporte fornecido pela

entidade coordenadora do concurso aos tutores, como os kits de tecnologia e treinamento por exemplo, eram repassados para os estudantes através de e-mails.

Ao final de cada iteração o tutor avaliava as equipes e pontuava seguindo os critérios de avaliação contidos no edital. Após a última iteração a equipe com maior pontuação foi declarada vencedora e então premiada. As dificuldades enfrentadas por equipes e tutores durante esta fase foram as seguintes:

- ❖ Cada equipe depositava seu projeto em um ambiente distinto, dificultando avaliação dos tutores, devido eles terem que acessar cada um desses ambientes para acompanhar o andamento das atividades;
- ❖ As informações e os problemas eram replicados para cada equipe, visto que os meios de comunicação existentes eram descentralizados (por exemplo, página do concurso, página da disciplina, lista de e-mails, repositório da equipe A, da equipe B etc);
- ❖ As equipes, no início do concurso não dominavam as tecnologias propostas no modelo arquitetural retardando o cumprimento do cronograma;
- ❖ A aderência da interface do componente com o padrão proposto no kit tecnológico.

No concurso em nível regional, após as equipes terem sido inscritas no concurso, o líder de cada equipe realizou o cadastro do projeto no FLOApp (como descrito na seção 4.1), para então ter acesso a todas as ferramentas disponibilizadas pelo mesmo.

A comissão do concurso dividiu o conjunto de casos de uso para ser implementados em duas iterações. Assim cada equipe deveria disponibilizar um *release* parcial ao fim da primeira iteração e um *release* final, contento todos os requisitos levantados, na iteração final.

Na primeira iteração, as equipes tiveram seu primeiro contato com um ambiente cooperativo. Então foi necessário disponibilizar um passo a passo sobre como cada equipe deveria proceder para a realização das atividades, como por exemplo, disponibilizar um *release* e fazer o *upload* dos artefatos. Já na segunda iteração, as equipes estavam familiarizadas com o ambiente e detinha o conhecimento prévio de quais informações deveriam ser disponibilizadas para o comitê organizador do concurso.

Cada projeto possui uma lista de discussão interna, acessada apenas pela equipe envolvida, é ainda através desta que o tutor pôde orientar seus alunos e também avaliar

o desempenho dos mesmos. Todavia, os problemas referentes ao domínio da aplicação e de forma geral aqueles envolvendo todas as equipes, tutores e CN, são tratados em listas de discussões gerais e fóruns. Segundo afirma Enman (2003) este fato não corresponde à realidade das comunidades de Software Livre existente, visto que estas não possuem um cliente específico e as contribuições são feitas de acordo com a necessidade da comunidade.

Com o objetivo de assegurar a qualidade dos casos de uso desenvolvidos, as equipes realizaram testes de aceitação automáticos, para tanto foi utilizado o plugin FLOAppTest (COITINHO et. all,2007), a ferramenta de teste de aceitação disponibilizada pelo FLOApp. Este *plugin* tem como objetivo recuperar do repositório um release do projeto e gerar um relatório de testes baseado num script que é criado de forma automática pelo usuário. Esse tipo de teste também visa certificar a implementação dos requisitos definidos na fase de análise.

Os problemas ocorridos durante esta fase foram:

- ❖ Devido aos critérios de avaliação do concurso, além do *release* liberado pelas equipes também deveria ser disponibilizados pacotes contendo nenhuma informação referente à equipe desenvolvedora. Contudo, o FLOApp não apresentou problemas de carga devido ao dobro dos dados terem sido armazenados.
- ❖ O *release* disponibilizado por cada equipe tinha seu tamanho superior a 16MB. Entretanto, o repositório utilizado pelo FLOApp possui uma limitação no tamanho máximo de cada pacote, sendo este limite de 2MB. Tornando a atividade de *upload* cansativa para as equipes como também para o comitê do concurso, que realizava o *download* de cada pacote, em seguida eram agrupados novamente para a avaliação. O tamanho máximo para *upload* não é um atributo fixo podendo portanto ser configurável, embora, como estava ocorrendo o concurso nenhuma mudança foi realizada no ambiente.

5.3 AVALIAÇÃO DOS RESULTADOS

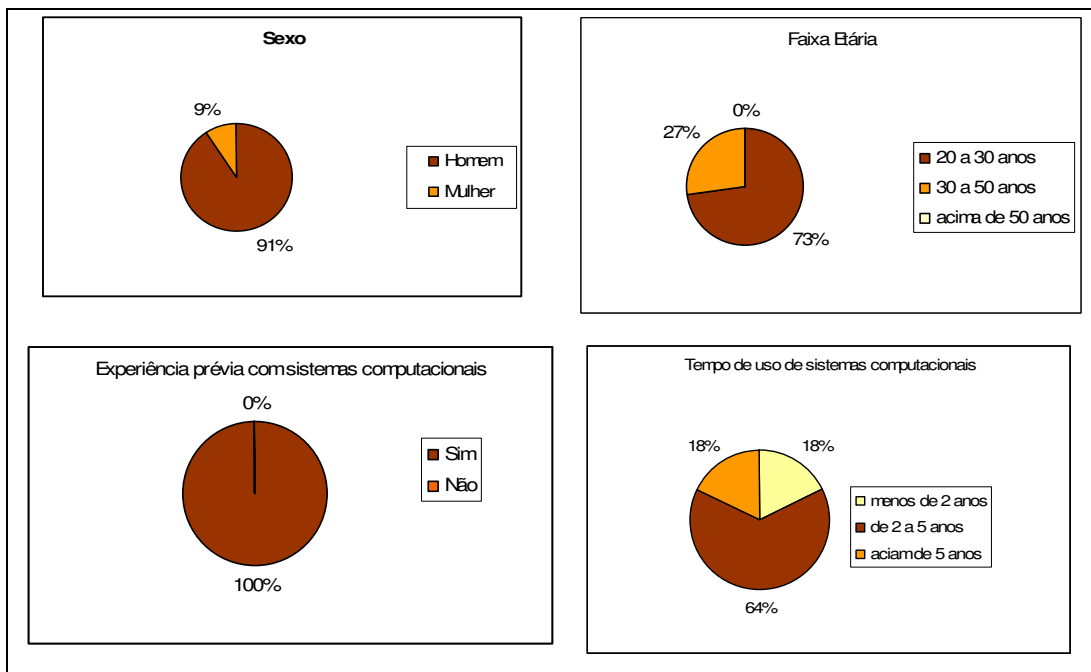
Para avaliar o uso do FLOApp pela comunidade técnica e de negócio durante a realização do concurso, foi utilizada a técnica de avaliação através da sondagem da satisfação subjetiva. Esta técnica consiste na análise das respostas dos usuários

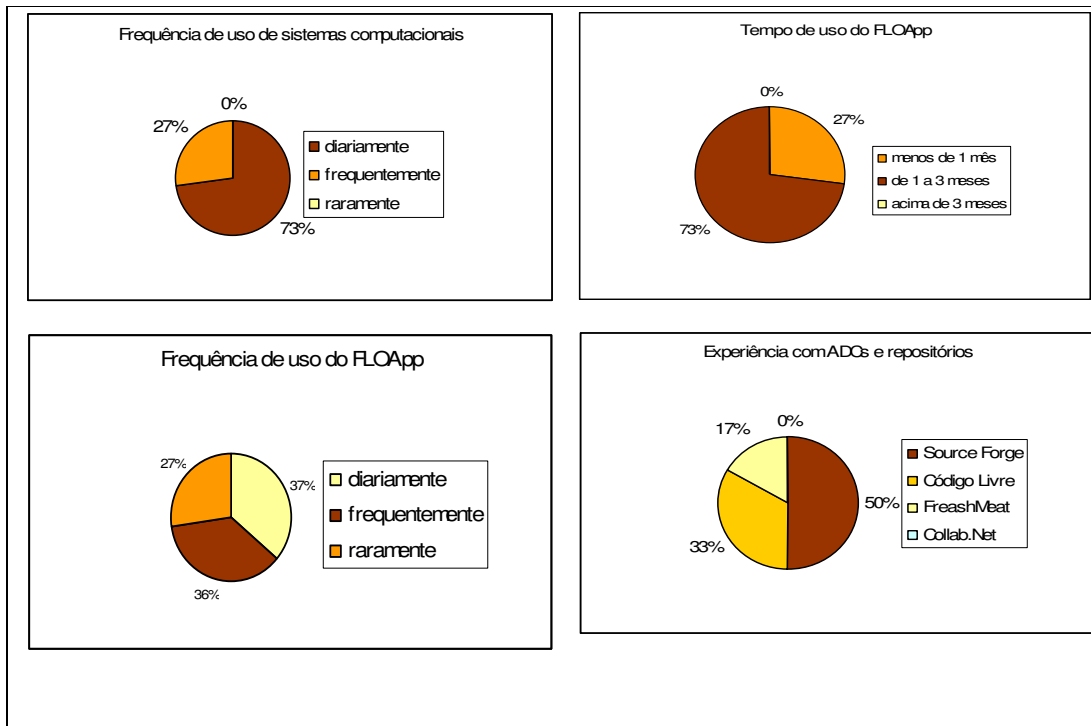
participantes do estudo de caso a um conjunto de questões relacionadas ao perfil do usuário e à satisfação do usuário durante a interação com o produto. Para a aplicação da técnica, foram utilizados, respectivamente, os questionários *USer (User Sketcher)* e *USE (User Satisfaction Enquirer)* do *Webquest* – uma ferramenta *web* configurável para o delineamento do perfil e a sondagem subjetiva do usuário (OLIVEIRA,2004).

A análise das respostas dos usuários participantes do estudo de caso aos itens presentes nos questionários *USer* e *USE* permite delinear o perfil dos usuários recrutados e sondar o grau de satisfação dos mesmos ao interagirem com o FLOApp. Os questionários são apresentados nos Anexos A e B respectivamente.

5.3.1 ANÁLISE DO QUESTIONÁRIO USER

Das comunidades envolvidas no concurso, foram recrutados oito participantes da comunidade técnica e três da comunidade de negócio para o levantamento do perfil do usuário do FLOApp, apresentado no Quadro 26:





Quadro 8 - Delineamento do perfil do usuário

5.3.2 ANÁLISE DO QUESTIONÁRIO USE

O questionário USE foi composto por 21 itens que abordaram aspectos relacionados à funcionalidade, usabilidade (facilidade de aprendizado e uso), situações de erro e travamentos, ajuda e satisfação de uso.

No tocante às funcionalidades, de forma geral, a avaliação do uso do FLOApp como ambiente de suporte ao desenvolvimento de componentes livres foi positiva, onde 63% usuários classificaram o item como fácil e 9% classificou como muito fácil.

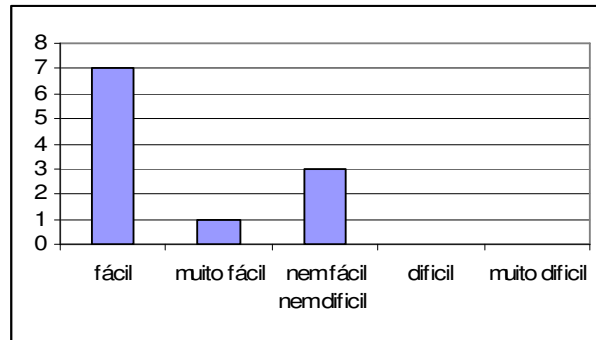


Figura 21 - Avaliação geral do uso do FLOApp.

De forma mais específica, a avaliação acima pode ser dividida na análise dos itens referentes às principais atividades do FLOApp, sejam essas: requisição de demandas, rastreamento por componentes, criação de projetos, liberação de *releases* e *download* de componentes.

A requisição de demandas foi avaliada positivamente pela comunidade de negócio, 66% consideraram a atividade como fácil e um 34% deles classificou como nem fácil nem difícil.

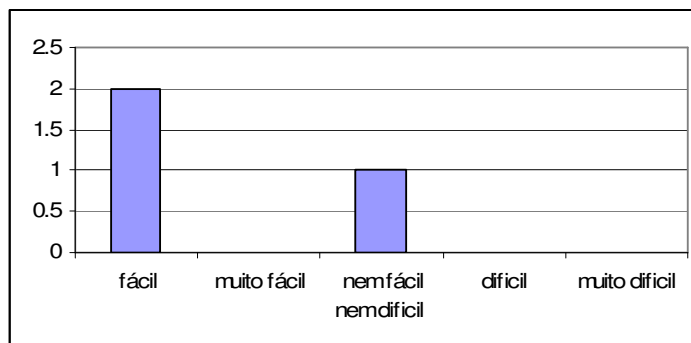


Figura 22 - Cadastro de demandas.

A avaliação para o rastreamento de componentes apresentou-se positiva pelos usuários do FLOApp ao considerar a realização da atividade como muito fácil 72% e fácil 28%.

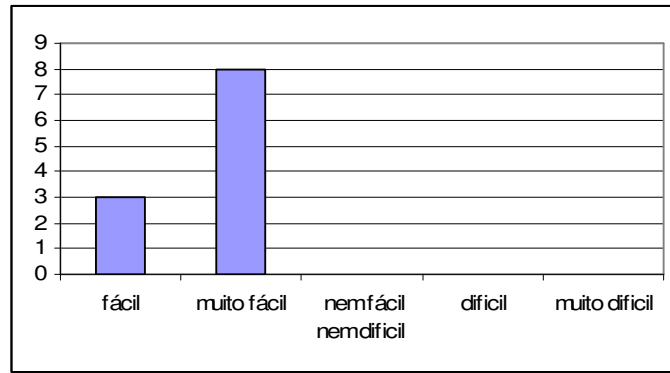


Figura 23 - Rastreamento de componentes.

A avaliação para a criação de projetos foi avaliada pela comunidade técnica como fácil 50%, nem fácil nem difícil 25% e difícil 25%.

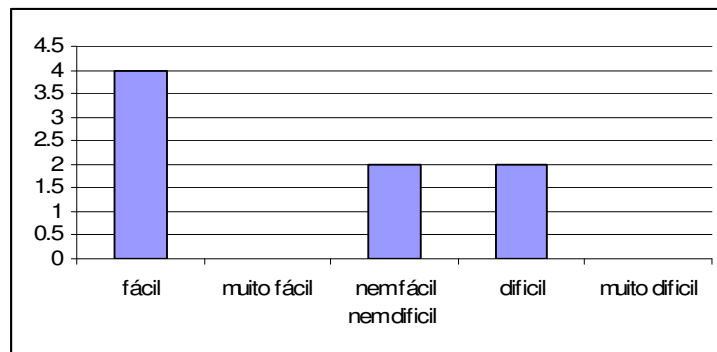


Figura 24 - Cadastrar novo projeto.

O *download* de componentes apresentou-se positiva pelos usuários do FLOApp ao considerar a realização da atividade como muito fácil 27% e fácil 73%.

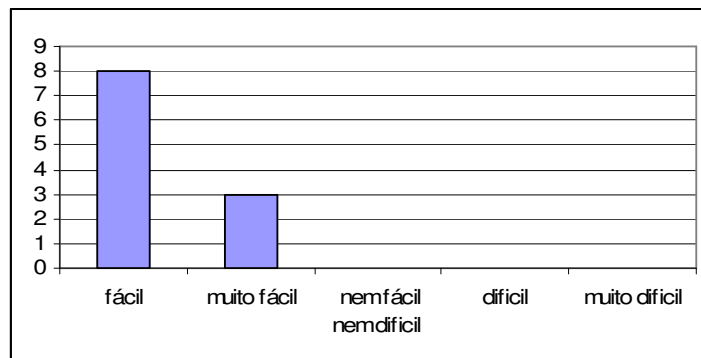


Figura 25 - Download de componentes.

Do ponto de vista da usabilidade, aspectos relacionados ao tempo de aprendizado; à facilidade de navegação e, à satisfação com a execução das atividades foram contemplados.

Com relação ao investimento de muito tempo para aprender a utilizar o FLOApp, revelou que 72% usuários discordaram (parcial e totalmente).

Outro aspecto, diz respeito à satisfação com a execução das atividades. Para esse item de avaliação 63% usuários demonstraram satisfação ao conseguir executar as tarefas de modo direto ao usar o FLOApp.

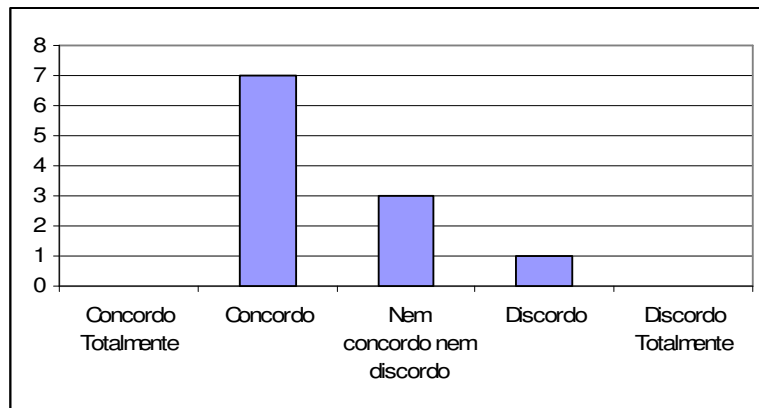


Figura 26 - Satisfação ao executar as atividades usando o FLOApp.

Para situações de erro e travamento, o FLOApp não apresentou nenhuma situação de travamento. Para recuperação de situações de erro, 54% usuários se mostraram satisfeitos. No entanto, com relação à compreensão de mensagens de erro, os usuários de FLOApp 72% consideraram as mensagens de erros difíceis de compreender.

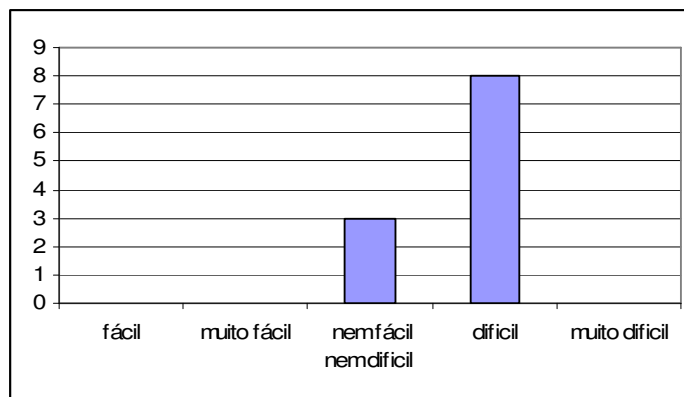


Figura 27 - Compreensão das mensagens de erro exibidas pelo FLOApp.

Analisando os aspectos relacionados à satisfação de uso, de modo geral todos os usuários demonstraram-se satisfeitos em utilizar o FLOApp, como mostrado na Figura 38.

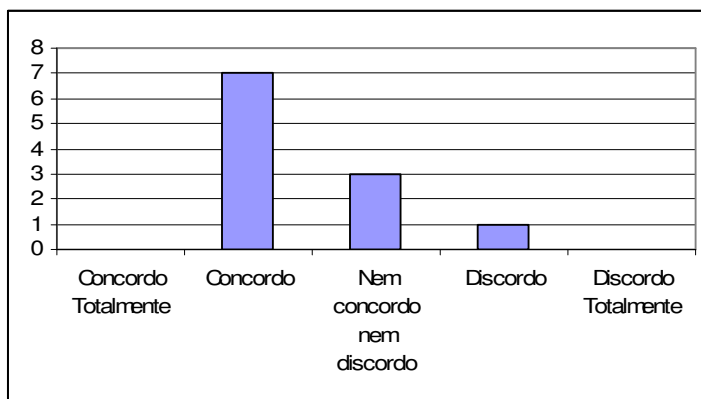


Figura 28 - Satisfação com o uso do FLOApp.

5.3.3 ÍNDICE DE SATISFAÇÃO SUBJETIVA

As respostas fornecidas pelos usuários ao questionário USE, utilizado para a avaliação do uso do FLOApp na execução do estudo de caso, são processadas pela ferramenta *Webquest* que fornece como resultado um índice de satisfação subjetiva de um dado universo amostral relativa a um produto considerado, no caso FLOApp.

O Quadro 26 contém a escala utilizada pelo USE para a valoração do índice de satisfação subjetiva.

INTERVALO	DESCRIÇÃO
0,67 a 1,00	Satisfação Máxima
0,33 a 0,66	Bastante Satisfeito
0,01 a 0,32	Satisfeito
0,00	Neutro
-0,01 a -0,32	Insatisfeito
-0,33 a -0,66	Bastante Insatisfeito
-0,67 a -1,00	Insatisfação Máxima

Quadro 9 - Análise do indicador de satisfação subjetiva do USE. (OLIVEIRA,2004)

O índice de satisfação subjetiva obtido neste experimento para o FLOApp foi -

0.312, o que indica que o universo amostral considerado mostrou-se **SATISFEITO** com o produto avaliado.

Embora o FLOApp não tenha sido avaliado pelo universo amostral de forma a atingir um dos índices máximos de satisfação, com base nos indicadores quantitativos coletados durante a realização o estudo de caso, nos próprios indicadores de satisfação subjetiva do usuário aos quais o experimento chegou, é possível afirmar que o uso do FLOApp obteve uma avaliação positiva.

5.4 CONSIDERAÇÕES FINAIS

Este capítulo apresentou um estudo de caso implementado em duas edições do concurso de Software Livre envolvendo estudantes, professores e prefeitura. A primeira foi uma edição reduzida envolvendo apenas alunos da disciplina de Engenharia de Software da UFCG. Neste concurso nenhum ambiente de suporte ao desenvolvimento foi utilizado pelas comunidades envolvidas. Foram então observadas as dificuldades enfrentadas e as soluções tomadas naquele momento. De modo geral as principais dificuldades enfrentadas foram:

- ❖ Perda de informações;
- ❖ Falta de interação entre os participantes;
- ❖ Falta de comunicação;
- ❖ Repositório dos artefatos descentralizado;
- ❖ Concentração de atividades com o tutor;
- ❖ Acesso ao cliente;

A segunda edição do concurso envolveu seis instituições de ensino, com a participação de mais de cinquenta estudantes e três participantes da comunidade de negócio (um prefeito e dois secretários de finanças e administração). Nesta edição a comunidade em geral contou com o apoio do FLOApp como ambiente de interação e desenvolvimento cooperativo. Durante a execução do concurso, os problemas citados acima com o uso do FLOApp foram resolvidos como detalhado abaixo:

- ❖ Perda de informações;

O FLOApp é um ambiente integrado de desenvolvimento, todas as atividades de gestão de projetos, comunicação e desenvolvimento são armazenadas no

próprio repositório do ambiente. Após o usuário ter efetuado o log in, o ambiente apresenta todas as últimas informações envolvendo este usuário.

❖ Falta de interação entre os participantes;

O FLOApp promove um ambiente de interação, no qual as equipes podem colaborar seja através de fórum, listas de discussões, reportando melhorias ou falhas, ou contribuindo com o próprio código.

❖ Falta de comunicação;

Com a utilização do FLOApp, tanto a comunidade técnica como a de negócio puderam fazer uso de listas de discussões e fóruns para tirar dúvidas e dar sugestões sobre o processo do concurso, o componente desenvolvido etc.

❖ Repositório dos artefatos descentralizado;

Como mencionado anteriormente, o FLOApp dispõe de um repositório, que permite a gestão de vários projetos, promovendo aos tutores e a entidade coordenadora do concurso a concentração das atividades de avaliação e acompanhamento em um único ambiente.

❖ Concentração de atividades com o tutor;

❖ Acesso ao cliente;

Os estudantes, na primeira edição do concurso, tiveram dificuldades de acesso ao cliente sobrecarregando as atividades do tutor. Na segunda edição com o uso do FLOApp, qual possui uma visão específica para a comunidade de negócio, mesmo o cliente (prefeitos e secretários) não estando acessível, especialistas no domínio do aplicação (gestão de imóveis) puderam interagir com a comunidade técnica através do fórum de discussão e em conjunto com a entidade coordenadora do concurso foi elaborado o documento de caso de uso, facilitando portanto, a compreensão por parte dos estudantes.

Entretanto, outras atividades executadas através do FLOApp exigiram uma maior compreensão e esforço por parte da comunidade. As principais dificuldades foram:

❖ Liberação de versões superiores a 2MB;

❖ Dificuldade na compreensão das mensagens de erro;

❖ A burocratização no cadastro de usuário.

Contudo, confrontando os resultados obtidos com os objetivos deste trabalho, podemos avaliar a contribuição como positiva. Observando que foi possível inserir a comunidade de negócio no processo de desenvolvimento de componentes livres, permitindo com que esta possa solicitar e priorizar suas necessidades e interagir diretamente com a comunidade responsável pela produção dessas demandas.

Por outro lado, a comunidade técnica pôde utilizar um ambiente com o qual já possui familiaridade, sabendo que o FLOApp utiliza o repositório *GForge*. E diferentemente do *Source Forge*, que tem seu código fechado, a própria CT pode, caso sinta necessidade, desenvolver e adicionar um ferramenta ao ambiente ou adicionar uma já produzida pela comunidade. Além disso o FLOApp, diferente dos ADCs e repositórios analisados disponibiliza uma ferramenta para testes de aceitação, podendo esta ser utilizada ou não pela comunidade técnica.

Entretanto, não foi possível gerar todas as métricas planejadas inicialmente, como as de reuso, quais deveriam auxiliar a CN na tomada de decisão. Todavia, com a utilização mais expressiva do FLOApp será facilmente inserida esta contribuição. E ainda, pela avaliação realizada através da sondagem subjetiva do usuário, de modo geral, todos se mostraram satisfeitos com os serviços disponibilizados.

CAPÍTULO 6

CONCLUSÕES E TRABALHOS FUTUROS

O conjunto de práticas adotadas pelas comunidades de Software Livre existentes tem espelhado várias empresas a alterarem sua maneira de desenvolver software. O modelo de desenvolvimento e manutenção baseado no compartilhamento do código-fonte do software tem sido apresentado como mais eficiente e econômico. Sua superioridade está na versatilidade e capacidade de agregar muito mais inteligência coletiva ao processo de elaboração e de melhoria de uma solução. Quanto maior for a comunidade que analisa uma versão de um projeto, mais amplamente as falhas poderão ser encontrados e mais velozmente elas poderão ser corrigidas.

No entanto, observando os projetos de Software Livre existentes foi identificada uma grande diversidade de objetivos, formas de participação e técnicas de implementação. Ainda foi possível identificar um desafio comum a todos os projetos: a necessidade de criação e manutenção de um ambiente sociotécnico onde os participantes possam, cooperativamente, construir soluções para problemas de interesse mútuo.

Diante dessa visão, este trabalho apresentou um modelo de interação buscando atrair a comunidade de Software Livre, formada por estudantes universitários, empresas de TI e usuário final pertencente ao negócio próprio do domínio, para formar uma rede de colaboração gerando conhecimento, empreendedorismo local e informatizando aquele domínio de interesse.

Apresentamos o resumo das contribuições deste trabalho na seção 6.1 e o conjunto de trabalhos futuros identificados na seção 6.2.

6.1 RESUMO DAS CONTRIBUIÇÕES

De forma geral, as principais contribuições desta dissertação foram:

- ❖ Viabilizar a colaboração das comunidades envolvidas no processo de desenvolvimento de aplicações livres para o domínio do negócio.
- ❖ Detalhar do modelo de negócio Via Digital que pode ser aplicável a qualquer domínio e aplicação.
- ❖ Fornecer uma plataforma de suporte ao Concurso de Software Livre Empreendedor, realizado pelo Via Digital.
- ❖ Desenvolver o FLOApp, orientado a uma arquitetura de plugins, simplificando a adição de novas ferramentas ao ambiente.

No Quadro 28 é apresentado o estudo abordado no Capítulo 2 comparando os existentes e agora com o FLOApp.

	<i>Source Forge</i>	<i>GForge</i>	<i>Fresh Meat</i>	Código Livre	Collab .Net	FLOApp
Comunidade Técnica						
Rastreamento de falhas	x	x	x			x
Geração de código	x	x	x			x
Pacote de distribuição	x	x	x	x	x	x
Integração de código						
Teste (execução e cobertura)						x
Métricas de documentação						x
Métricas de reuso						
Qualificação			x			x
Controle de versão	x	x	x	x	x	x
Privacidade	x	x	x	x	x	x
Segurança	x	x	x	x	x	x
Comunidade de Negócio						
Rastreamento de componentes	x	x	x	x	x	x
Gestão de demandas						x

	<i>Source Forge</i>	<i>GForge</i>	<i>Fresh Meat</i>	Código Livre	Collab .Net	FLOApp
Comunidade de Negócio						
Indicadores de satisfação						X
FAQ (Dúvidas Frequentes)						

Quadro 10 - Diferenças entre os ambientes existentes e o proposto.

Logo, os resultados apresentados atingem os objetivos desta dissertação. Não obstante alguns pontos de melhoria que foram identificados ao longo da elaboração deste trabalho, que são apresentados na seção que segue.

6.2 TRABALHOS FUTUROS

O modelo de desenvolvimento de Software Livre apresenta-se como uma forte tendência no que diz respeito a métodos inovadores de desenvolvimento de software. Neste sentido, novas recursos tecnológicos se tornam necessários. Além disso, ao longo do nosso trabalho identificamos oportunidades de melhorias nos serviços oferecidos pelo FLOApp, incluindo:

- ❖ No âmbito do concurso nacional, englobar um número maior de participantes da comunidade de negócio e assim será possível obter melhores respostas das métricas fornecidas pelo FLOApp;
- ❖ Adicionar de ferramentas de integração ao FLOApp;
- ❖ Adicionar métricas de reuso;
- ❖ Disponibilizar uma FAQ para a comunidade de negócio;
- ❖ Melhorar a compreensão das mensagens de erro apresentadas.

REFERÊNCIAS BIBLIOGRÁFICAS

- (ALVES et.all,2006) A. M., Souza. A. D., Silva. L. P. e Silva, T. V. *Escritório de Gerenciamento de Projetos Múltiplos, Dispersos Geograficamente, No Âmbito do Programa Via Digital*. Anais do II Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES 2006). Vila Velha : PESC/COPPE/UFRJ, 2006. p. 1-12.
- (BIRSAN,2005) Birsan, D.. *On Plug-ins and Extensible Architectures* - 2005. ACM Queue, 3(2):40–46.
- (BOSTON,2002) The Boston Consulting Group. *The Boston Consulting Group Hacker Survey* - 2002. Disponível em: <http://www.bcg.com/opensource/BCGHackerSurveyOSCON24July02v073.pdf>.
- (BOOCH, 2002) Booch, Grady & Brown, Alan W. *Collaborative Development Environments*. Rational Software Corporation, 2002.
- (BROWN, 1996) Brown, A.W., K.C Wallnau. *Component-Based Software Engineering*. IEEE Computer Society Press. Pags. 7-15. 1996.
- (BSD,2007) BSD: Berkeley Software Distribution – Disponível em: <http://www.bsd.org/>.
- (CODLIVRE, 2007) Código Livre: O principal banco de códigos livres ordenados de Brasil. Disponível em: <http://www.codigolivre.com.br/>
- (COLLAB, 2007) Collab.Net: Ambiente Colaborativo de Desenvolvimento de Software. Disponível em: <http://www.collab.net/>

- (COITINHO,2007) Coitinho et. all. *Web-based tool for automatic acceptance test execution and scripting for programmers and customers*. Proceedings EATIS 2007 - Euro American Conference on Telematics and Information Systems. Faro, Portugal. ISBN # 978-1-59593-598-4.
- (CROWSTON,2004) Crowston, Kevin & et. All. *Effective Work Practices for Software Engineering: Free/Libre Open Source Software Development*. Syracuse University School of Information Studies. 2004.
- (CRUZ,2003) Cruz, T. *Sistemas, Métodos & Processos: administrando organizações por meio de processos de negócios*. São Paulo: Atlas. 2003.
- (DPROJECT,2007) Ferramenta para gerência de atividades – DotProject. Disponível em: <http://www.dotproject.net/>
- (EASYP,2007) Processo de Desenvolvimento easY Process. Disponível em: <http://www.dsc.ufcg.edu.br/~yp>.
- (FELLER,2000) Feller, Joseph & Fitzgerald, Brian. A framework analysis of the open source development paradigm. University College Cork. Ireland.2000.
- (FIELDING,1999) Fielding, R. T. Shared leadership in the Apache Project. *Communications of the ACM*, v. 42, n. 4, p. 42-43, Abril 1999.
- (FOURNIER,2001) Fournier, Roger. *InfoWorld*, Ed. Março de 2001.
- (FRESHMEAT,2007) FreshMeat: Web's largest index of Unix – Disponível em: <http://freshmeat.net>.

- (FSWF,2005) The Free Software Foundation. *What is Free Software?* - Disponível em <http://www.gnu.org/philosophy/free-sw.html>. 2005.
- (GAMMA, 2000) Gamma, E., Helm, R., Johnson, R., Vlissides, J. *Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos*. Editora Bookman, Porto Alegre, RS, 2000.
- (GFORGE,2007) Framework Gforge – Disponível em: <http://gforge.org>
- (GIURI,2002) Giuri, Paola & et. All. *Open source software: from open science to new marketing models*. Università di Camerino and Scuola Superiore Sant’Anna, Pisa. 2002.
- (GOLDEN,2005) Golden, Bernard. *Succeeding with open source*. Ed. Addison-Wesley Information Technology Series.2005.
- (GPL,2007) GPL: GNU General Public License - Disponível em: <http://www.gnu.org/copyleft/gpl.html>.2007.
- (GREENBERG,2003) GREENBERG, Gerald. & et. All. *Open Source: Moving into the Enterprise*. CUTTER Information LCC.2003.
- (HAMMER,1998) Hammer, M., Station, S. *How process enterprises really work*. Havard Business Review, v.77, n.6, p.108-118, Nov/Dez.
- (HIPPEL,2005) Hippel, Eric von. *Open Source Software Projects as User Innovation Networks*. Perspectives on Free and Open source Software. FELLER, Joseph [et all]. Massachusetts Institute Technology, 2005. p. 267-278.

- (IBGE,2001) IBGE, Censo 2001, Instituto Brasileiro de Geografia e Estatística (IBGE - The Brazilian Institute of Geography and Statistics). Available at www.ibge.gov.br , 2001.
- (KRISHNA,2005) Krishnamurthy, Sandeep. *An analysis of Open Source Business Models*. Perspectives on Free and Open source Software. FELLER, Joseph [et all]. Massachusetts Institute Technology, 2005. p. 279-296.
- (LERNER,2005) Lerner, Josh & Triole, Jean. Economic Perspectives on Open Source. Perspectives on Free and Open source Software. FELLER, Joseph [et all]. Massachusetts Institute Technology, 2005. p. 47-78.
- (LIMA,2005) Lima, C. A. *Práticas para Gerência e Desenvolvimento de Projetos de Software Livre*. Universidade Federal de Campina Grande.2005
- (LGPL,2007) LGPL: GNU Lesser Public License - Disponível em: <http://www.gnu.org/licenses/lgpl.html>.2007.
- (LUCCA,2007) De Lucca, J. E., Savi, R. *Produção compartilhada de software para governos municipais*. Anais do VIII Fórum Internacional de Software Livre (fisl), 2007. Porto Alegre, RS.
- (MARGERISON,2004) Margerison, Dean. *Collaborative Development Environments*. Dekam Consultancy. 2004
- (MCKUSICK,1999) Mckusick, M. K. *Twenty years of Berkeley Unix*. In: Open Sources .Sebastopol: O'Reilly and Associates, 1st. ed., 1999. p. 31-46.

- (MOURA,2006) MOURA, A. et. All. *Open Source Software in Small City Governments and the Promotion of Regional Entrepreneurship*. eChallenges Proceedings 2006. Barcelona, Outubro 2006.
- (MPL,2007) MPL: Mozilla Public License - Disponível em: <http://www.mozilla.org/MPL/>.2007.
- (MUNIZ,2007) Muniz, A., Oliveira, J. A. *OSS Factory: Development Model based at OSS Practices*. Proceedings ICEIS 2007 - International Conference on Enterprise Information Systems. Funchal, Portugal.
- (OLIVEIRA,2004) Oliveira, J. E. L. *Processo, que processo?*. RAE – Revista de Administração de Empresas. São Paulo. v40 – p.8-19. Out/Dez.
- (RAYMOND,1999) Raymond, E. S. *The Cathedral and The Bazaar*. In: *The Cathedral and The Bazaar*. Sebastopol: O'Reilly and Associates, 1st. ed., 1999. p. 5-78.
- (ROZANSKI,2005) Rozanski, N. & Woods, E. *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives*. Addison Wesley. 2005.
- (SENTANIN,2004) Sentanin, O. F. *Gestão por processo em uma empresa de pesquisa e desenvolvimento: objetivo estratégico de um modelo de gestão*. Escola de Engenharia de São Carlos da Universidade de São Paulo. 2004.
- (SFORGE,2005) Source Forge – Disponível em: <http://sourceforge.net>
- (SOFTEX,2004) SOFTEX, Report - *Projeto Aplicação de Software Livre em Prefeituras*. SOFTEX Association, 2004.

- (SOLISC,2005) Congresso Catarinense de Software Livre. Florianópolis, SC.
Disponível em: <http://www.solisc.org.br/index.php>
- (SOMMER,2003) Sommerville, S. I. *Engenharia de Software*, Addison Wesley,
6ª edição, 2003.
- (SOUZA,2000) Souza, H. A. *Gestão Educacional: um instrumento de
avaliação para o segmento educação*. Secretaria de Estado de
Educação do Rio de Janeiro/ UNIABEU, 2000.
- (STALLMAN,1999) Stallman, R. M. *The GNU Operating System and the Free
Software Movement*. In: Open Sources. Sebastopol: O'Reilly
and Associates, 1st. ed., 1999. p. 53-70.
- (VALINUX,2007) VA Linux Systems. Disponível em:
<http://www.valinux.co.jp/en/>
- (WIKIPEDIA,2007) Wikipedia – The Free Encyclopedia. Disponível em:
<http://www.wikipedia.org>.

APÊNDICE: DETALHAMENTO DO MODELO DE NEGÓCIO EM PROCESSOS

1 EMPREENDEDORISMO

Este macro processo visa levantar expectativas locais da comunidade técnica. Fazer e aprovar projetos de captação de recursos para o provimento de concursos. Planejar e implementar ações de criação de empregos e de novos negócios na região em torno das atividades de prestação e qualificação de serviços (personalização, integração, treinamento etc) usando Software Livre.

1.1 Planejamento	
Descrição:	Este processo visa à definição de estratégias, ações (inclusive a definição do <i>portfólio</i> de produtos e serviços adequados) e recursos para o alcance de sucesso nos mercados alvos.
KPI:	<ul style="list-style-type: none">❖ % de metas cumpridas❖ Custo e tempo de planejamento❖ % variação dos KPIs
Ferramentas:	<ul style="list-style-type: none">❖ Planilha❖ Editor de Texto❖ Gestão de Tarefas
Atores:	Entidade sem fins lucrativos (EFL), comunidade técnica (CT), comunidade de negócios (CN)

Quadro 11 - Detalhamento do Processo Planejamento.

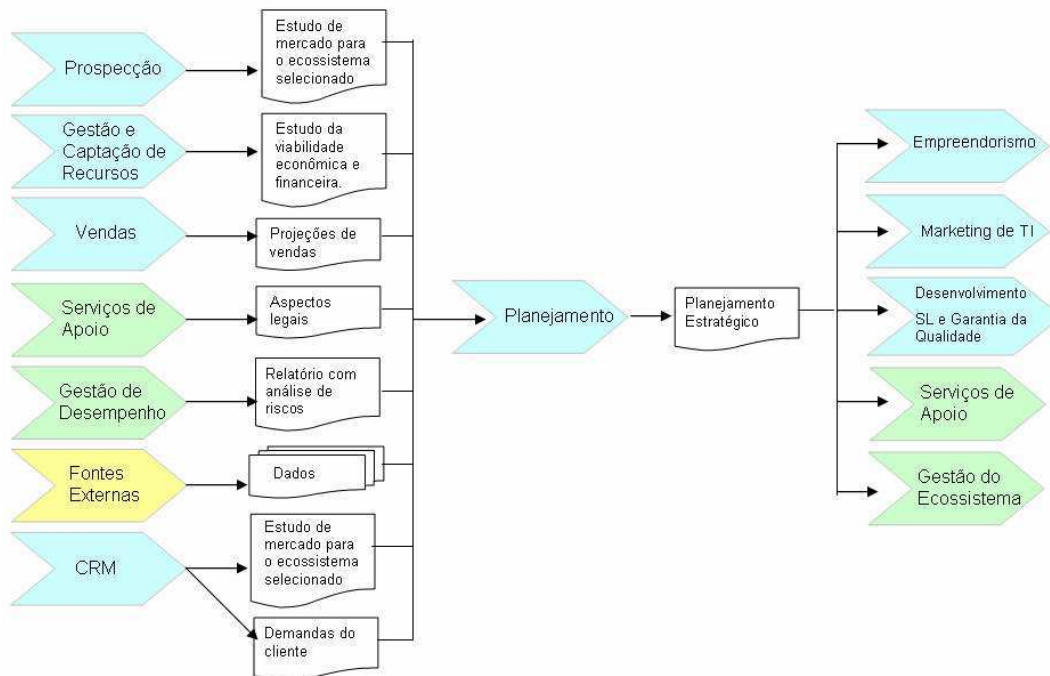


Figura 29 - Fluxograma do Processo Planejamento.

1.2 Gerência de Concurso	
Descrição:	Este processo visa o planejamento, convocação, execução, avaliação, suporte e premiação de contribuições em aspectos técnicos e mercadológicos para os mercados alvos.
KPI:	❖ Número de participantes dividido pelo custo de realização do concurso.
Ferramentas:	<ul style="list-style-type: none"> ❖ Site do concurso ❖ Repositório ❖ Canais de comunicação ❖ Mídia impressa ❖ Kits de acompanhamento
Atores:	Entidade sem fins lucrativos (EFL), comunidade técnica (CT), comunidade de negócios (CN)

Quadro 12 - Detalhamento do Processo Gerência de Concurso.

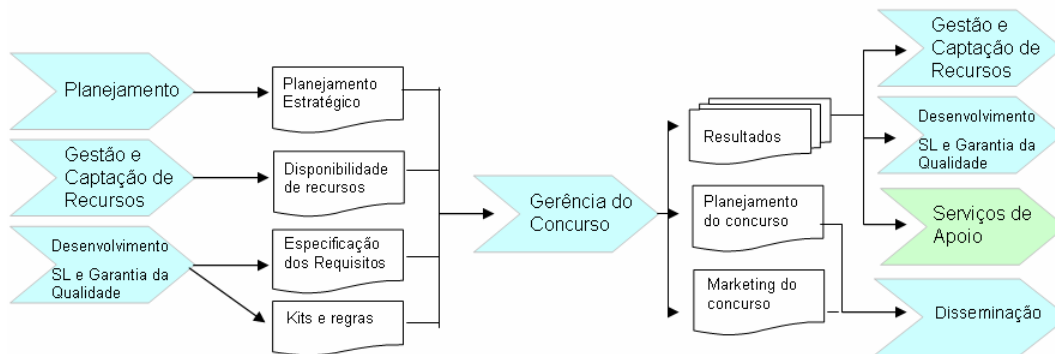


Figura 30 - Fluxograma do processo Gerência do Concurso.

1.3 Gestão e Captação de Recursos	
Descrição:	Este processo visa o planejamento, negociação, captação e prestação de contas de recursos econômicos (a exemplo de espaços para incubação) e financeiros internos e externos ao ecossistema, inclusive sua regularização e controle.
KPI:	❖ Recursos obtidos sobre os custos totais.
Ferramentas:	<ul style="list-style-type: none"> ❖ Planilha ❖ Editor de texto ❖ Canais de comunicação
Atores:	Entidade sem fins lucrativos (EFL)

Quadro 13 - Detalhamento do Processo Gestão e Captação de Recursos.

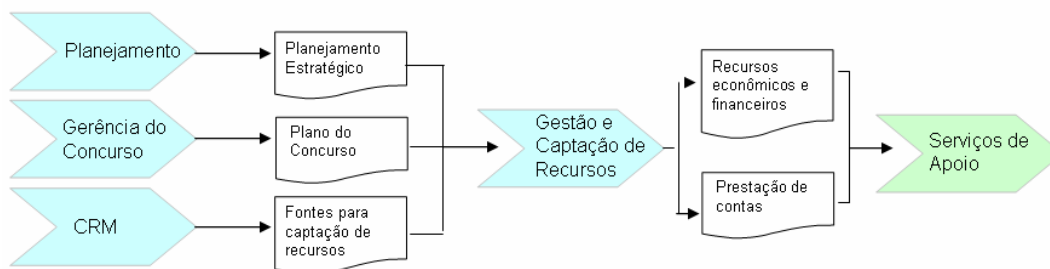


Figura 31 - Fluxograma do processo Gestão e Captação de Recursos.

1.4 CRM (Customer Relationship Management)	
Descrição:	Este processo visa o cadastro, classificação e acompanhamento do relacionamento com os clientes internos e externos.
KPI:	❖ Frequência de vendas para clientes existentes.
Ferramentas:	❖ Redes de relacionamento ❖ Sistema de CRM ❖ Máquinas de busca
Atores:	Entidade sem fins lucrativos (EFL), comunidade técnica (CT), comunidade de negócios (CN)

Quadro 14 - Detalhamento do Processo CRM.

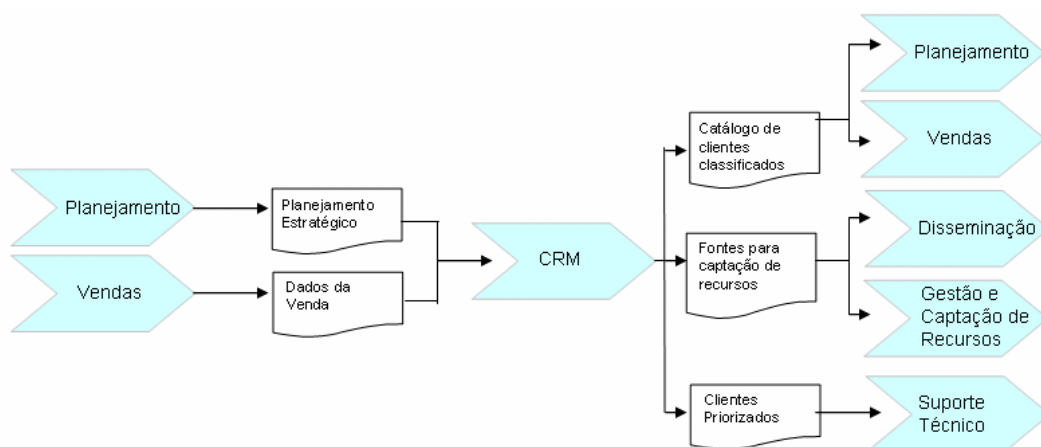


Figura 32 - Fluxograma do processo CRM.

2 MARKETING EM TI

É o conjunto de processos que tratam das interações do ecossistema e a organização gestora e asseguram a satisfação dos clientes e sua auto-sustentação.

2.1 Prospecção	
Descrição:	Este processo visa o estudo de mercados alvos e identificação de oportunidades, riscos e necessidades, inclusive com mapeamento dos fatores críticos de sucesso.
KPI:	❖ % de oportunidades identificadas e exploradas.
Ferramentas:	❖ Planilha ❖ Sistema de Busca ❖ Sistema de CRM
Atores:	Entidade sem fins lucrativos (EFL), comunidade técnica (CT), comunidade de negócios (CN)

Quadro 15 - Detalhamento do Processo Prospecção.

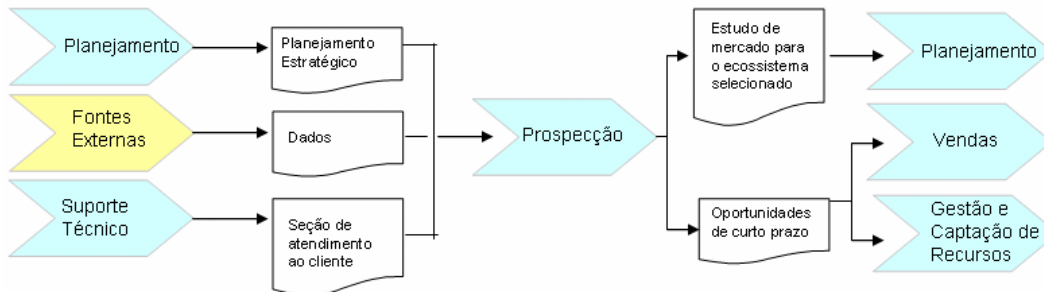


Figura 33 - Fluxograma do processo Prospecção.

2.2 Disseminação	
Descrição:	Este processo visa à elaboração e execução de campanha e material de divulgação para diversas mídias junto aos clientes internos e externos.
KPI:	❖ % de familiaridade sobre a ação no ecossistema antes e depois.
Ferramentas:	❖ Mídia impressa ❖ Sítio de acompanhamento ❖ Canais de comunicação
Atores:	Entidade sem fins lucrativos (EFL)

Quadro 16 - Detalhamento do Processo Disseminação.

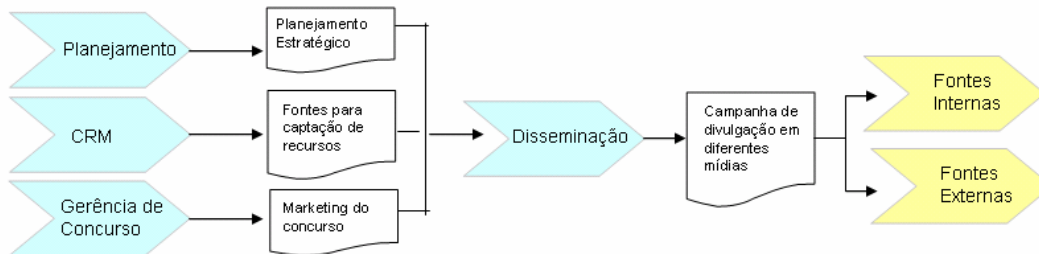


Figura 34 - Fluxograma do processo Disseminação.

2.3 Vendas	
Descrição:	Este processo visa à comercialização e contratação de produtos e serviços junto aos mercados alvos.
KPI:	❖ O volume de vendas sobre o número de pessoas associadas ao Via Digital.
Ferramentas:	❖ Planilha ❖ Editor de texto
Atores:	Entidade sem fins lucrativos (EFL), comunidade técnica (CT), comunidade de negócios (CN)

Quadro 17 - Detalhamento do Processo de Vendas.

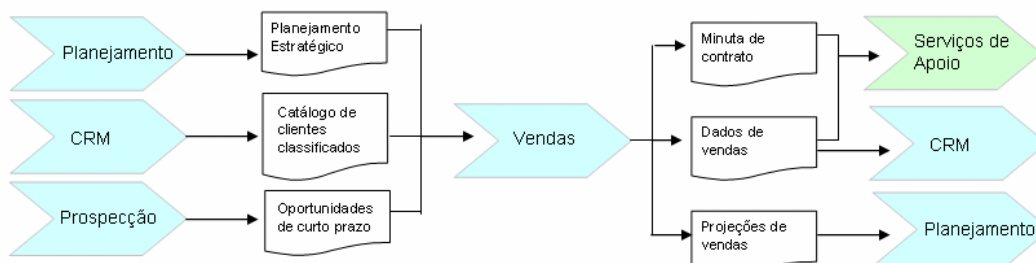


Figura 35 - Fluxograma do processo Vendas.

3 SERVIÇOS DE APOIO

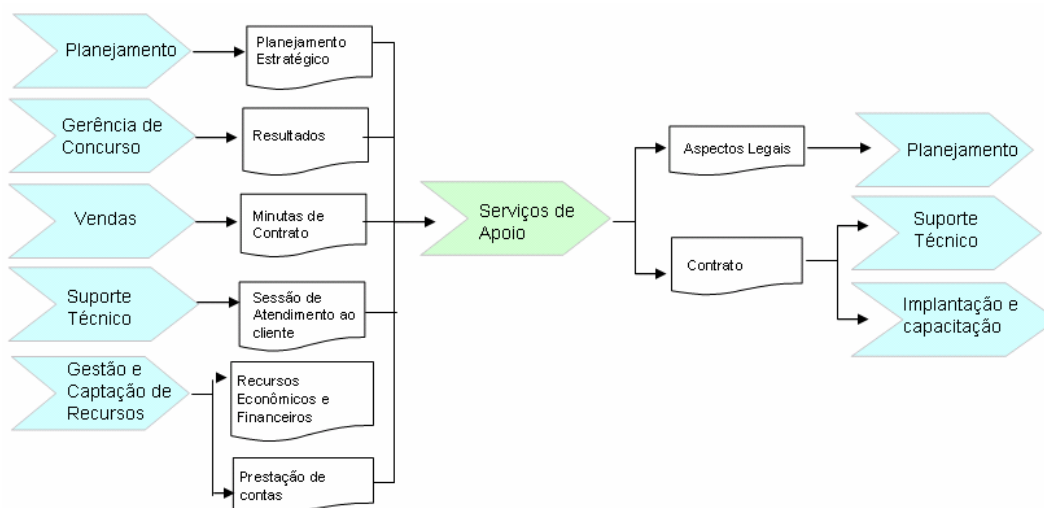


Figura 36 - Fluxograma do processo Serviços de Apoio.

3.1 Serviços Administrativos Auxiliares

Descrição: Abrangem as atividades de transporte, recebimento, tramitação e expedição de documentos, reprografia, controle de ligações telefônicas, conservação, limpeza e segurança física de bens e instalações.

Quadro 18 - Detalhamento do Processo Serviços Administrativos Auxiliares.

3.2 Serviços Jurídicos

Descrição: Este processo tem por finalidade tratar das políticas, diretrizes e procedimentos relativos às orientações jurídicas preventivas e corretivas, bem como às medidas de defesa judicial e administrativa da instituição e seus parceiros, junto às pessoas jurídicas de direito público e privado e perante as diversas instâncias do poder judiciário.

Quadro 19 - Detalhamento do Processo Serviços Jurídicos.

3.3 Gestão de Recursos Financeiros	
Descrição:	Este processo envolve a direção de atividades econômico-financeiras, incluindo planejamento econômico-financeiro, orçamento, contabilidade, contas a receber, contas a pagar, tesouraria, controle de custos, controle de recursos, controle da dívida, controle patrimonial e gestão de riscos e seguros.

Quadro 20 - Detalhamento do Processo Gestão de Recursos Financeiros.

4 GESTÃO DO ECOSISTEMA

Atividades de monitoração, comunicação, acompanhamento e avaliação de desempenho dos processos de negócios do ecossistema de modo a apoiar os resultados para atendimento dos objetivos estratégicos.

4.1 Gestão do Desempenho	
Descrição:	É o processo focado na monitoração, comunicação, avaliação acompanhamento e melhoria do desempenho de processos de negócio de forma a apoiar os resultados do negócio na instituição, contribuindo assim para a consecução de objetivos estratégicos.
KPI:	<ul style="list-style-type: none"> ❖ Número de processos melhorados (por período de avaliação); ❖ Redução do risco total associado ao plano estratégico; ❖ % variação do KPI acima; ❖ KPIs globais.
Ferramentas:	<ul style="list-style-type: none"> ❖ Planilha ❖ Editor de texto
Atores:	Entidade sem fins lucrativos (EFL)

Quadro 21 - Detalhamento do Processo Gestão do Desempenho.

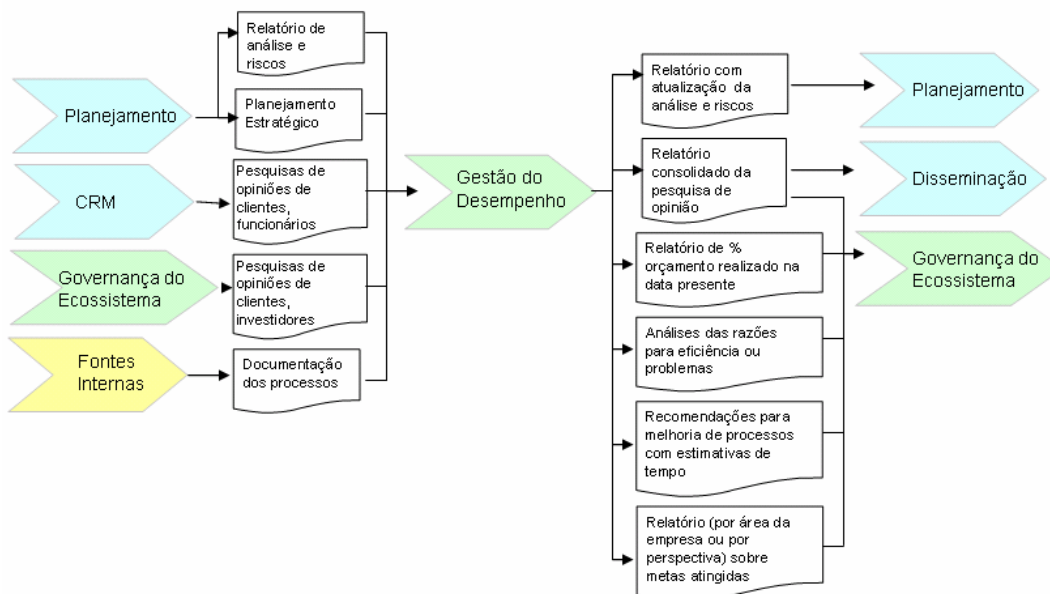


Figura 37 - Fluxograma do processo Gestão do Desempenho.

4.2 Governança do Ecosistema	
Descrição:	A governança do ecossistema preocupa-se em garantir que os procedimentos conduzam os dirigentes da instituição oss.org a agir – com possibilidade de rastreamento de responsabilidade – em prol dos interesses da própria instituição, do ecossistema, dos parceiros e de seus investidores.
KPI:	<ul style="list-style-type: none"> ❖ Número de recomendações acatadas pela diretoria e % variação deste KPI; ❖ Custo x tempo da realização da avaliação e % variação deste KPI.
Ferramentas:	<ul style="list-style-type: none"> ❖ Planilha ❖ Editor de texto
Atores:	Entidade sem fins lucrativos (EFL)

Quadro 22 - Detalhamento do Processo Governança do Ecosistema.

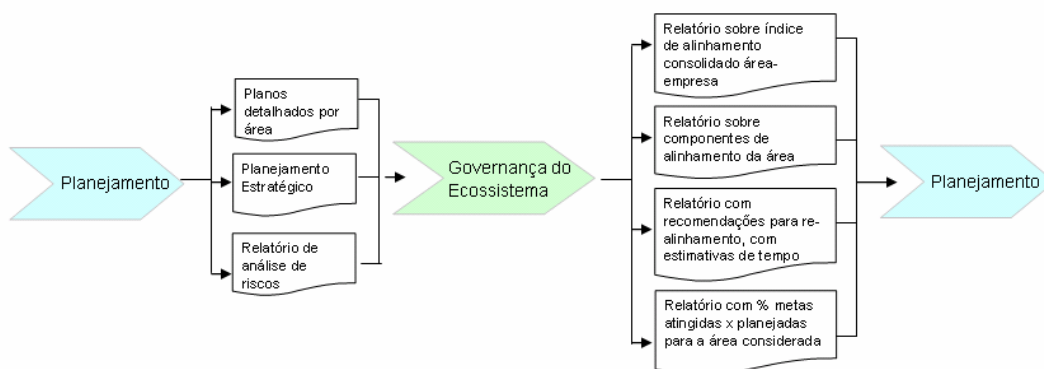


Figura 38 - Fluxograma do processo Governança do Ecosistema.

5 DESENVOLVIMENTO SL E GARANTIA DA QUALIDADE

Este macro processo prevê a criação de um ambiente auto-sustentável que auxilie a promoção do empreendedorismo e também ofereça suporte ao desenvolvimento de componentes livres e serviço para um domínio de aplicação específico, assegurando e certificando a qualidade de ambos.

A partir da descrição do fluxo geral são apresentados abaixo os sub-processos correspondentes. Estes compreendem todas as atividades do processo de desenvolvimento de um componente, caracterizando as comunidades envolvidas e as ferramentas que devem ser fornecidas pelo ambiente.

5.1 Engenharia de Requisitos	
Descrição:	Este processo visa o levantamento e análise de requisitos funcionais e não funcionais junto ao cliente.
KPI:	❖ % de requisitos especificados vs. Implementados.
Ferramentas:	❖ CASE (Rational) ❖ DOORS(Quality System and Software Ltda) ❖ UML ❖ Editor de texto
Atores:	Comunidade técnica (CT)

Quadro 23 - Detalhamento do Processo Engenharia de Requisitos.

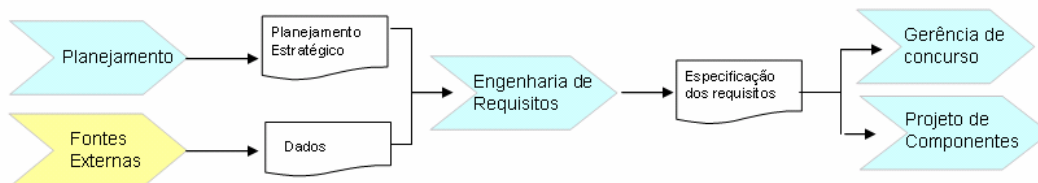


Figura 39 - Fluxograma do processo Engenharia de Requisitos.

5.2 Projeto de Componentes	
Descrição:	Este processo visa à elaboração e validação do projeto arquitetural de componentes, inclusive com mapeamento de componentes existentes e/ou a serem customizados.
KPI:	❖ Aprovação do líder do projeto; ❖ Número de mudanças no projeto.
Ferramentas:	❖ UML ❖ Editor de texto ❖ Editor gráfico ❖ Modelagem da Tarefa
Atores:	Comunidade técnica (CT)

Quadro 24 - Detalhamento do Processo Projeto de Componentes.

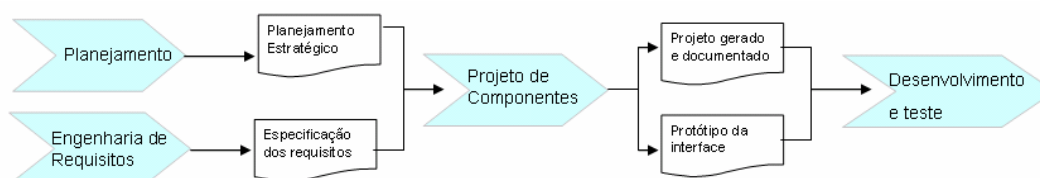


Figura 40 - Fluxograma do Processo Projeto de Componentes.

5.3 Desenvolvimento e Testes	
Descrição:	Este processo visa à codificação e testes de componentes, inclusive com validação junto ao cliente.
KPI:	❖ Número de <i>bugs</i> encontrados por linha de código; ❖ Número de programadores envolvidos (maior melhor devido SL).
Ferramentas:	❖ IDE (Ambiente de Desenvolvimento Integrado) ❖ Gestão de mudanças ❖ Integração contínua ❖ SGBD (Sistema de Gerenciamento de Banco de Dados) ❖ Rastreamento de falhas e/ou melhorias ❖ Teste funcional/unitário/performance/carga ❖ Canais de comunicação ❖ Documentação ❖ Gestão de tarefas/atividades
Atores:	Comunidade técnica (CT), Comunidade de negócios (CN)

Quadro 25 - Detalhamento do Processo Desenvolvimento e Testes.

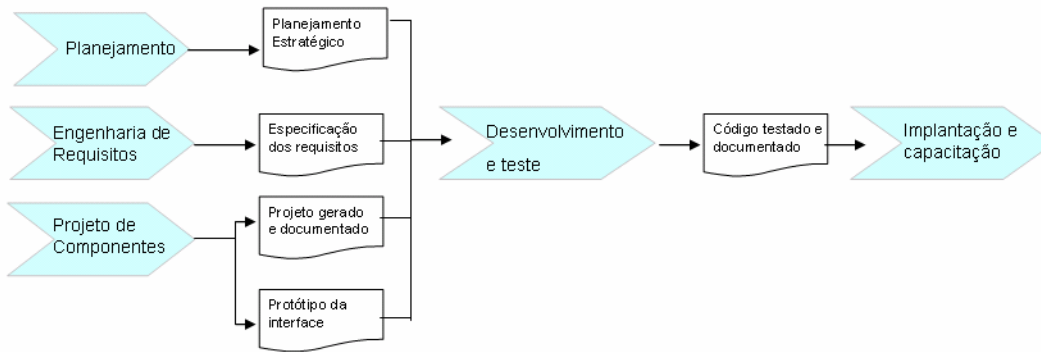


Figura 41 - Fluxograma do Processo Desenvolvimento e Testes.

5.4 Implantação e Capacitação	
Descrição:	Este processo visa o planejamento, empacotamento (artefatos e recursos) e instalação dos componentes junto ao cliente e a capacitação dos usuários finais.
KPI:	❖ Índice de satisfação do cliente.
Ferramentas:	<ul style="list-style-type: none"> ❖ Instalador automático ❖ Serviço de backup ❖ Editor de texto ❖ Editor de apresentação
Atores:	Comunidade de negócios (CN)

Quadro 26 - Detalhamento do Processo Implantação e Capacitação.

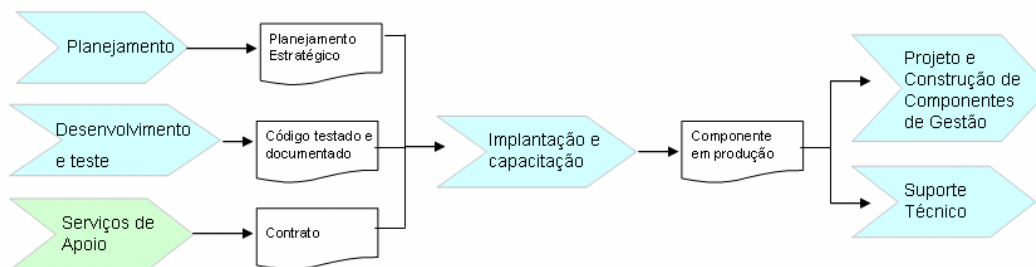


Figura 42 - Fluxograma do processo Implantação e Capacitação.

5.5 Projeto e Construção de Componentes de Gestão	
Descrição:	Este processo visa à elaboração, desenvolvimento e implantação de componentes para gestão do projeto (site, plataforma de desenvolvimento, gestão de configuração, documentação de artefatos, análise de custos).
KPI:	❖ Índice de satisfação da comunidade técnica.
Ferramentas:	❖ <i>Site</i> de acompanhamento ❖ Gestão de mudanças ❖ Editor de texto ❖ Planilha
Atores:	Entidade sem fins lucrativos (EFL)

Quadro 27 - Detalhamento do Processo Projeto e Construção de Componentes de Gestão.

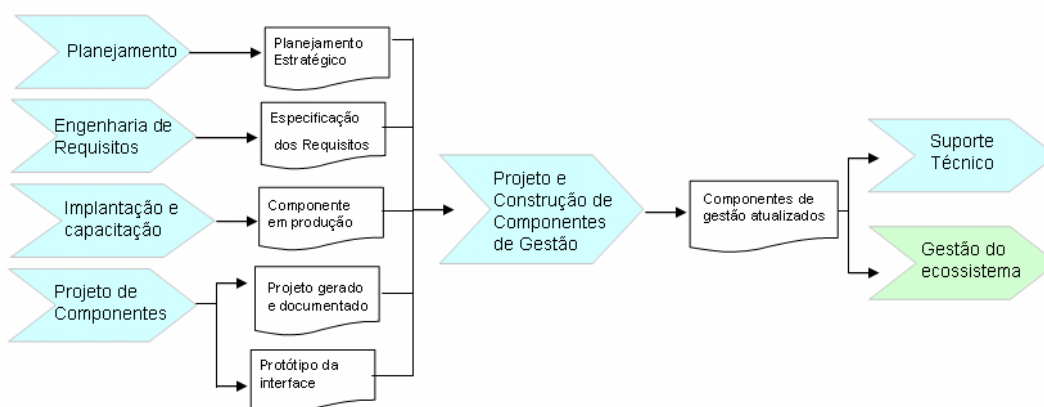


Figura 43 - Fluxograma do processo Projeto e Construção de Componentes de Gestão.

5.6 Suporte Técnico	
Descrição:	Este processo visa o planejamento, desenvolvimento e operação de <i>services desk</i> para os clientes internos e externos.
KPI:	❖ Duração da sessão (quanto maior pior); ❖ Índice de satisfação do cliente.
Ferramentas:	
Atores:	Entidade sem fins lucrativos (EFL), Comunidade de negócios (CN)

Quadro 28 - Detalhamento do Processo Suporte Técnico.

ANEXO A. DELINEAMENTO DO PERFIL DO USUÁRIO – QUESTIONÁRIO USER (USER SKETCHER)

- 1. Seu grau de instrução**
 Graduando Graduado Mestrando Mestre Outros
- 2. Você é do sexo**
 Masculino Feminino
- 3. Você pertence à faixa etária de**
 de 20 a 30 anos de 30 a 50 anos acima de 50 anos
- 4. Sua plataforma computacional é:**
 Windows 95/98/2000 Windows XP/Vista Linux
- 5. Você tem experiência prévia com sistemas computacionais?**
 Sim Não
- 6. Há quanto tempo você usa sistemas computacionais?**
 Menos de 1 ano Entre 2 a 5 anos Mais de 5 anos
- 7. Com que frequência você usa sistemas computacionais?**
 Raramente Frequentemente Diariamente
- 8. Há quanto tempo você usa o FLOApp?**
 Menos de 1 mês Entre 1 a 3 meses Mais de 3 meses
- 9. Com que frequência você utiliza o FLOApp?**
 Raramente Frequentemente Diariamente
- 10. Qual a forma de ajuda do FLOApp que você costuma utilizar mais frequentemente?**
 Fórum Lista de discussão Manual
- 11. Você tem experiência com ambientes colaborativos e/ ou repositórios existentes?**
 Sim Não
- 12. Quais desses ambientes colaborativos e/ou repositórios você já utilizou?**
 Source Forge FreshMeat Código Livre Collab.Net

ANEXO B. SONDAGEM SUBJETIVA DA SATISFAÇÃO DO USUÁRIO - QUESTIONÁRIO USE (USER SATISFACTION ENQUIRER)

- 1. O uso do FLOApp na realização das atividades de interesse da comunidade**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 2. A comunicação com o FLOApp (terminologia, linguagem, realimentação da informação e ações em geral) é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 3. A localização das ferramentas associadas às atividades da comunidade é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 4. A visualização das instruções e advertências apresentadas pelo FLOApp é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 5. A compreensão das instruções e advertências apresentadas pelo FLOApp**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 6. Navegação pelo fluxo de atividades do FLOApp é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 7. Compreensão dos termos e solicitações apresentadas pelo FLOApp é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 8. Recuperação de situações de erro ao utilizar o ambiente é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 9. A recuperação de situações de travamento ao utilizar o ambiente é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 10. A compreensão das mensagens de erro apresentadas pela FLOApp é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 11. A localização e acesso ao manual de ajuda disponibilizado pelo ambiente**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 12. A compreensão das informações de interesse existentes no manual**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 13. A requisição de demandas (comunidade de negócio) é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 14. O rastreamento de componentes (comunidade técnica e de negócio) é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil
- 15. A criação de novos projetos (comunidade técnica) é**
() Muito Fácil () Fácil () Nem Fácil Nem Difícil () Difícil () Muito Difícil

16. O download de componentes (comunidade técnica e de negócio) é

Muito Fácil Fácil Nem Fácil Nem Difícil Difícil Muito Difícil

17. As respostas do ambiente às minhas ações são muito lentas

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente

18. Consigo executar as atividades de modo direto ao usar o ambiente

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente

19. Acho que o ambiente atende plenamente às minhas necessidades

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente

20. Perco muito tempo tentando aprender a usar o ambiente

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente

21. Consigo fazer exatamente o que desejo com os recursos oferecidos pelo ambiente

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente

22. O ambiente apresenta tantas funcionalidades que às vezes acho que nunca aprenderei ou terei necessidade de usá-las todas

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente

23. Sinto-me receoso em alguns momentos ao usar o ambiente

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente

24. Não acho que as informações da ajuda são suficientemente eficazes para tirar minhas dúvidas

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente

25. Em alguns momentos sinto-me frustrado com o modo como o ambiente executa atividades do meu interesse

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente

26. As informações relativas aos componentes são suficientes na tomada de decisão

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente

27. As informações relativas às demandas são suficientes na tomada de decisão

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente

28. Sinto-me satisfeito com as informações de interesse disponibilizadas

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente

29. De um modo geral, sinto-me satisfeito ao usar o ambiente

Concordo Totalmente Concordo Nem Concordo Nem Discordo

Discordo Discordo Totalmente